AD-772 508

ASPECTS OF MECHANICAL BEHAVIOR OF ROCK UNDER STATIC AND CYCLIC LOADING. PART A: MECHANICAL BEHAVIOR OR ROCK UNDER STATIC LOADING

Jesus E. Basas, et al

Wisconsin University

Prepared for:

Advanced Research Projects Agency
Bureau of Mining

April 1973

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

ASPECTS OF MECHANICAL BEHAVIOR OF ROCK UNDER
STATIC AND CYCLIC LOADING

PART A: MECHANICAL BEHAVIOR OF ROCK UNDER STATIC LOADING

Final Technical Report
April, 1973

by

R.W. Heins (Co-Principal Investigator
with B.C. Haimson)

Department of Metallurgical & Mineral Engineering
and the
Engineering Experiment Station
College of Engineering
The University of Wisconsin
Madison, Wisconsin 53706

ARPA Order No. 1579, Amendment 3
Program Code No. 2F10. Contract No. H0220041
Contract Period: March 1972 through April 1973
Total Amount of Contract: $50,000
Sponsored by ARPA and Monitored by U.S. Bureau of Mines

ic

AD 772508

3200.8 (Att 1 to Encl 1)
Mar 7, 66

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Engineering Experiment Station University of Wisconsin Madison, Wisconsin 53706	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE Aspects of Mechanical Benavior of Rock under Static and Cyclic Loading Part A - Mechanical Behavior of Rock under Static Loading	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Annual Technical Report	
5. AUTHOR(S) (First name, middle initial, last name) Jesus E. Basas, Robert W. Heins	
6. REPORT DATE April 1973	7a. TOTAL NO. OF PAGES 99 / 7b. NO. OF REFS 4
8a. CONTRACT OR GRANT NO. H0220041 b. PROJECT NO. ARPA Order No. 1579 Amendment 3 c. Program Code No. 2F10 d.	9a. ORIGINATOR'S REPORT NUMBER(S) 144-C745 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
10. DISTRIBUTION STATEMENT Distribution of this document is unlimited.	
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Washington, D.C. 20301

13. ABSTRACT

A finite element program based on both 2- and 3-dimensional element models is developed to simulate the Brazilian test. Rock isotropy, non-homogeneity, and anisotropy are taken into account in the analysis. The program has two failure criteria options: the old criterion described in the semi-annual report and a new criterion in which the elastic modulus across tension cracks is reduced to zero. Tests runs have shown that the new failure criterion yields far more realistic load-displacement and load-strain curves.

A new equation solving procedure, the sparse matrix technique, is used in the program making possible the solution of certain problems which could not be handled by the banded matrix technique because of computer storage limitations. To demonstrate the capabilities of the program, several test problems were run. The results are shown in Chapter 2 of this report.

DD FORM 1 NOV 65 1473

Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Brazilian test						
finite element method						
2- and 3-dimensional elements						
isotropy						
anisotropy						
non-homogeneity						
sparse matrix						
isoparametric elements						

Security Classification

PREFACE

This report covers the second year accomplishments in the research program entitled, "Mechanical Behavior of Rock Under Static Loading," R.W. Heins, Co-Principal Investigator. The program is Part A of the project entitled, "Aspects of Mechanical Behavior of Rock Under Static and Cyclic Loading" (Contract No. H0220041). Part B of the project is published in a separate volume.

ACKNOWLEDGEMENTS

The reported research was supported by ARPA and monitored by Dr. Syd S. Peng of the Twin Cities Mining Research Center, U.S. Bureau of Mines. His cooperation and that of Mr. Egons S. Podnieks are gratefully acknowledged.

The work reported here has been the principal responsibility of Dr. Jesus E. Basas who was a postdoctoral fellow on this project. The contribution of Mr. A. Hayatdavoudi on a laboratory investigation of size effect in the Brazilian test which was previously reported in a semiannual report should be noted.

SUMMARY

ASPECTS OF MECHANICAL BEHAVIOUR OF ROCK UNDER STATIC LOADING

PART A

Summary of Year's Work

Brazilian tests were carried out on dacite, Valder's limestone, and St. Cloud gray granodiorite to determine size effect on tensile strength. Curves showing variation of tensile strength with specimen dimensions and other findings constitute Chapter 1 of the semi-annual technical progress report[1].

In connection with the theoretical phase of the investigation, a 2-dimensional computer program which simulates the Brazilian test is developed. The program is based upon isoparametric finite elements which may be either isotropic or non-homogeneous and a failure criterion in which a fixed percentage of the stiffnesses of the failed elements is subtracted from the total stiffness during each loading cycle. The complete program listing and the results of several computer runs are part of the semi-annual report[1].

A second program is presented in this report. The program employs 2- and 3-dimensional isoparametric elements which exhibit anisotropy as well as isotropy and non-homogeneity. Two failure criteria are used: the old criterion described in the preceding paragraph and a new criterion in which the elastic modulus across tension cracks is reduced to zero. Based on few test runs made, the new failure criterion yields considerably superior and more realistic load-displacement and load-strain curves.

To demonstrate the capabilities of the program, several problems were run and the results are shown in Chapter 2. The use of a new equation solver, the sparse matrix technique, permitted the solution of certain problems which could not be handled by the banded matrix technique because of the bandwidth limitation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PART A

ASPECTS OF MECHANICAL BEHAVIOR OF ROCK UNDER STATIC AND CYCLIC LOADING

A FINITE ELEMENT MODEL OF ROCK FAILURE FROM THE BRAZILIAN TEST

Jesus E. Basas
Robert W. Heins

April, 1973

Department of Metallurgical & Mineral Engineering
The University of Wisconsin
Madison, Wisconsin 53706

CHAPTER 1

PROGRAMMING ROUTINES

1.1 Introduction

A finite element program for the prediction of rock fracture characteristics in Brazilian tests is presented herein. The program has all the capabilities of the previous 2-dimensional program (1) plus the following:

1. Availability of both 2- and 3-dimensional elements with isotropic, anisotropic, or non-homogeneous material property.

2. A failure criterion in which the elastic modulus across tension cracks is reduced to zero.

3. An equation solver in which only the non-zero elements of the stiffness matrix are involved in the elimination and back-substitution processes.

4. A nodal reordering scheme.

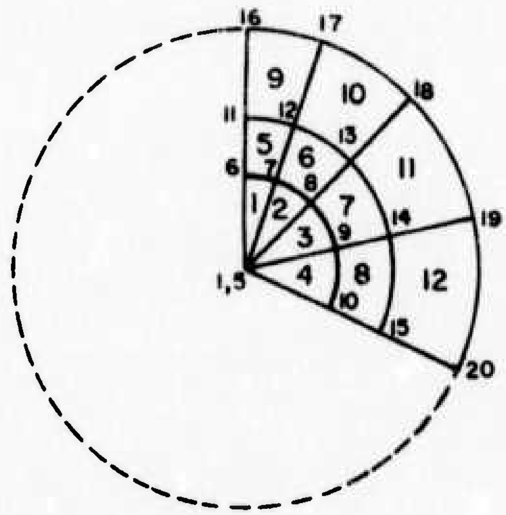
The theoretical considerations upon which the program is based have already been discussed in the semi-annual report (1) and are not repeated here. The program's major routines are described in the following sections.

1.2 Nodal Reordering

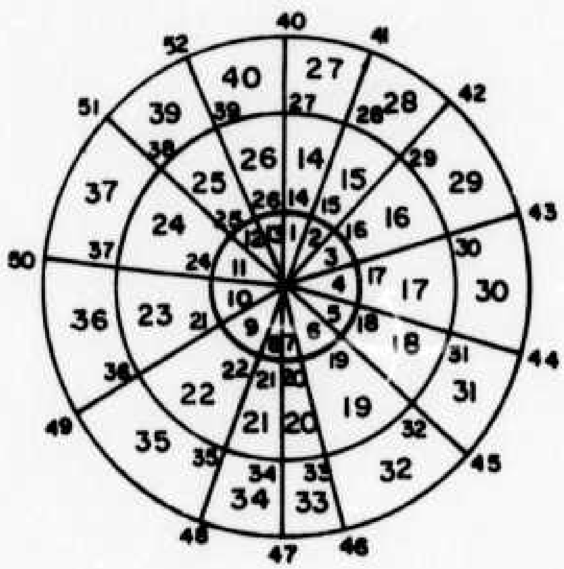
Among the different mathematical undertakings encountered in a finite element program, the solution of the governing linear equilibrium equations determines to a large extent the effectiveness of the program. When the first finite element programs were developed only small-size systems could be treated because of limited memory space for the storage of the global

stiffness matrix. With the introduction of external storage disks, it became possible to store only a small part of the stiffness matrix in computer memory at a time, the balance being in the disk, permitting the solution of considerably larger systems. The most widely used disk-aided equation solving procedure is the banded matrix technique developed in 1965 by E.L. Wilson of the University of California. Wilson's method takes advantage of the banded property of the stiffness matrix resulting in savings in execution time and storage requirement. Its only limitation is in the width of the band. Unfortunately, the maximum bandwidth that most digital computers allow is not large enough in many practical situations. In the context of the present research, such situations arise when the whole circular face of the Brazilian test specimen is considered in the analysis and/or when 3-dimensional problems are treated. In both of these cases, because of the closed geometry of the finite element mesh and the high nodal connectivity of the 3-dimensional elements, a low-numbered node becomes connected to a high-numbered node resulting in prohibitively large bandwidths. Nodal reordering schemes have been developed to minimize the bandwidth. The effectiveness of such schemes, however, is very limited.

An equation solver which stores and processes only the non-zero elements of the stiffness matrix is employed in the present program. To minimize the number of non-zero elements created during the elimination process, a reordering of the nodes is undertaken. The reordering concept used is based upon one described by Jensen and Parks (2). The complete reordering flow chart is shown in Figure 1.3. To start with, the nodes are assigned numbers in the manner shown in Figures 1.1 and 1.2. This allows for the automatic generation, within the program, of the nodes and their connections. The reordering of the nodes then starts. At each stage of the reordering process, the branch table of the last node (JLØC) entered in the

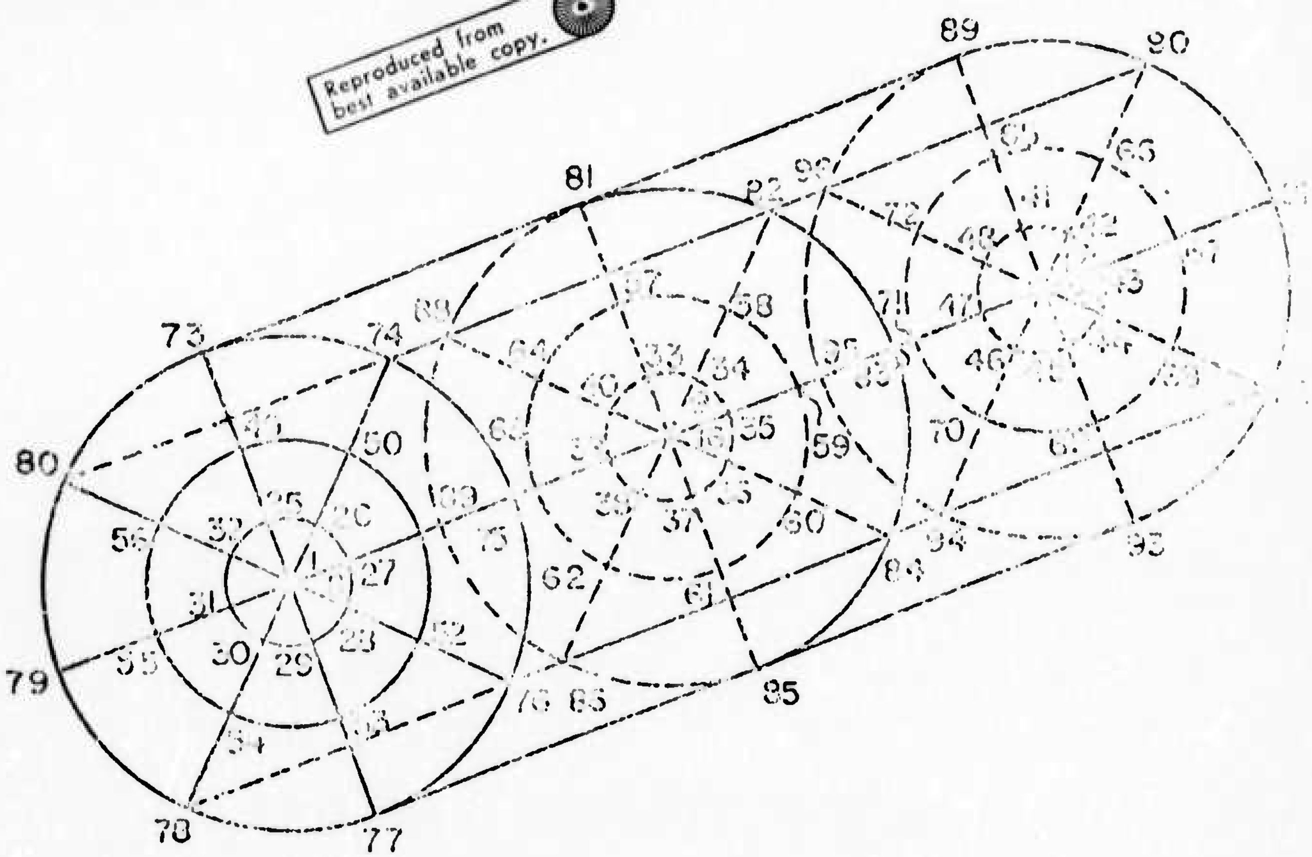


(a) Partial Disc



(b) Full Disc

Figure 1.1. Standard Scheme for Numbering Nodes and Elements of Two-dimensional Disc.



Note: Numbering of elements follows the same direction.

Figure 1.2 Numbering Scheme for Nodes in Three-dimensional Mesh

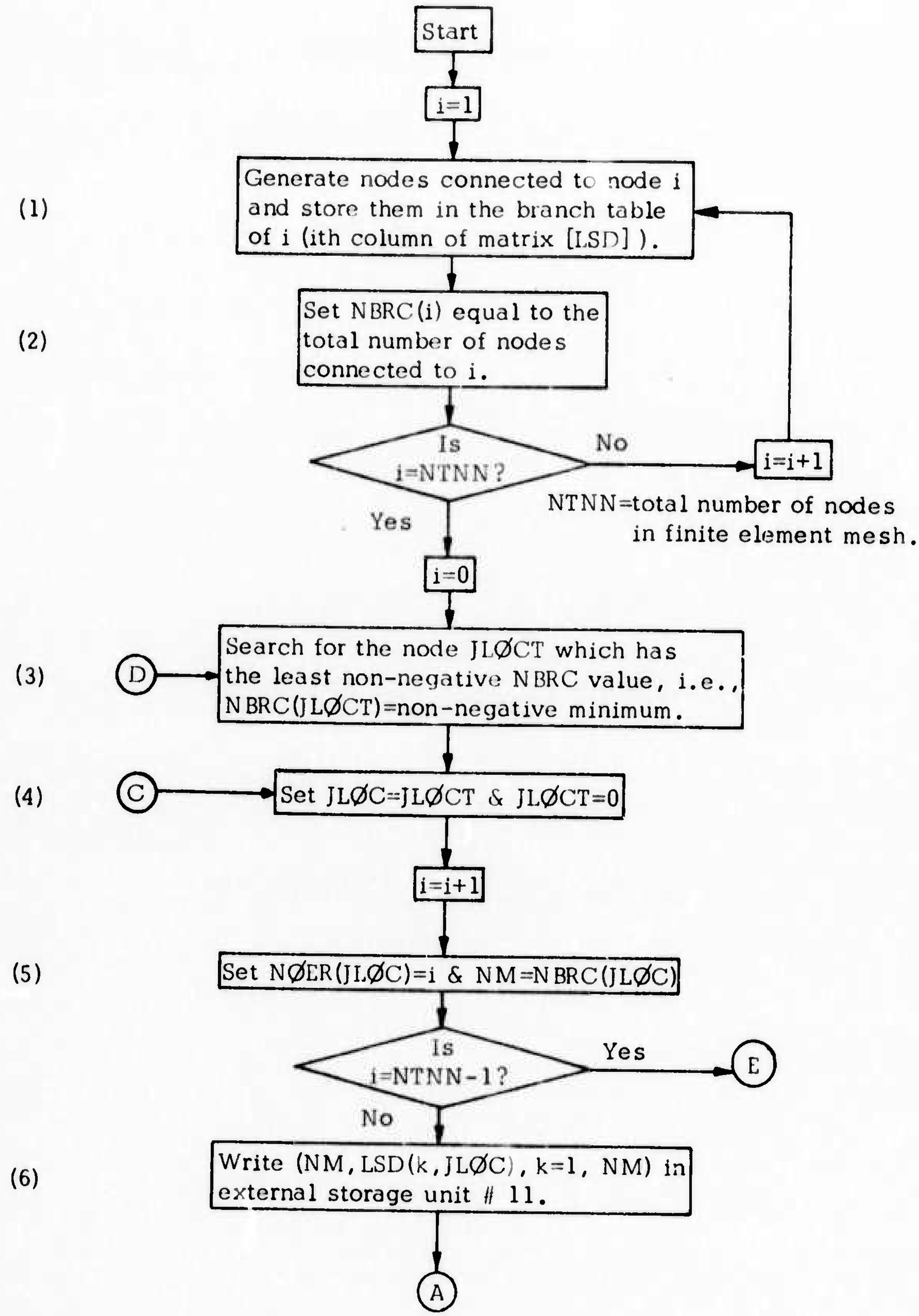


Figure 1.3. Flow Chart of Nodal Reordering Routine.

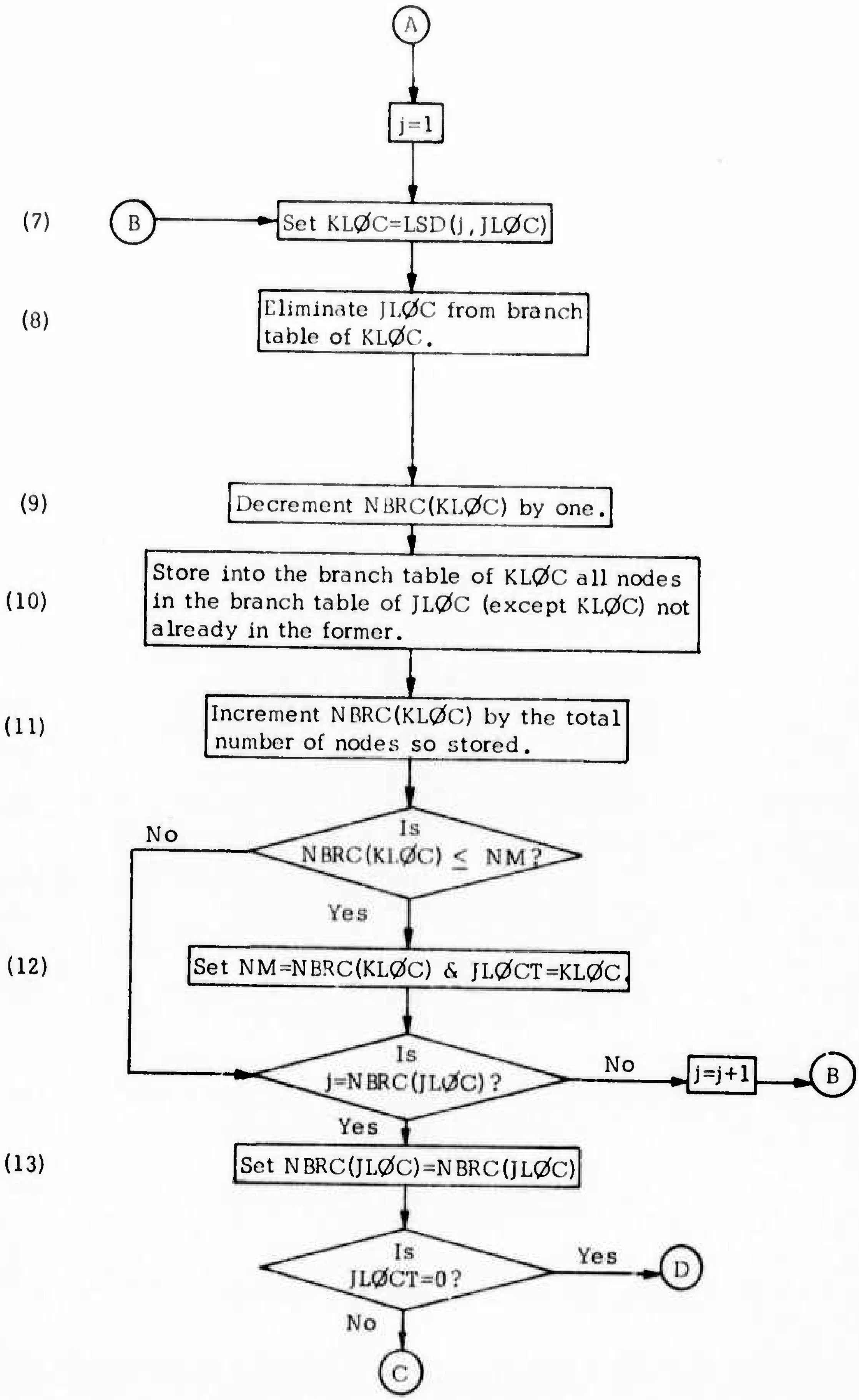


Figure 1.3 Continued.

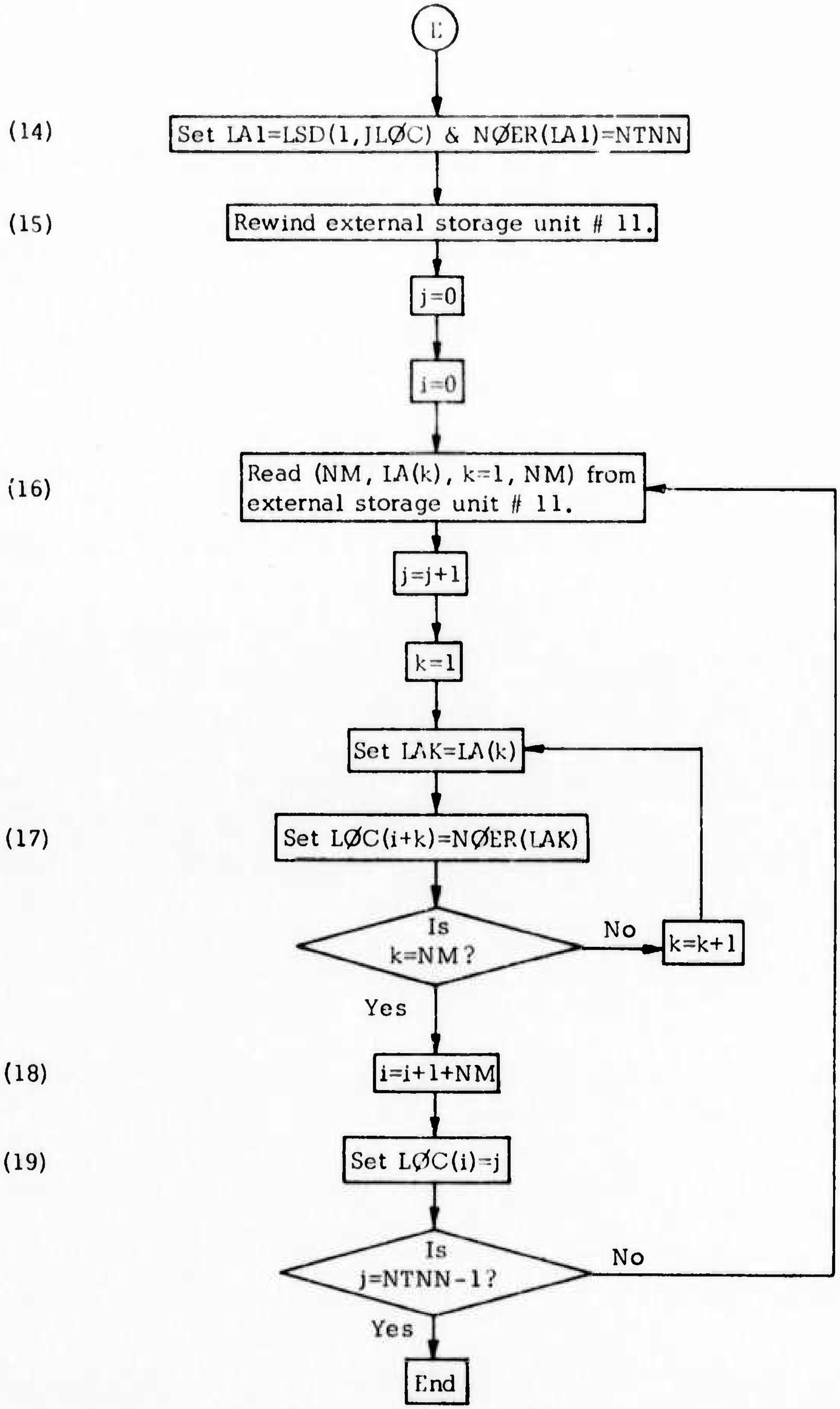


Figure 1.3 Continued.

reordering array* is scanned for the node with the fewest number of connections. If such node has equal or fewer connections than JLØC, then it is placed into the next available space of the reordering array. Otherwise, the node with the fewest number of connections from among those which remain to be renumbered is placed into the next available space of the reordering array. The process is repeated until all nodes have been renumbered. Whenever more than one node satisfy the aforementioned conditions, the node with the lowest number is selected. An example of the effect of reordering is illustrated in Figures 1.4 and 1.5. In Figure 1.4c no reordering is undertaken and 18 non-zero elements (represented by the letter c) are created during the elimination process. In the reordered version, Figure 1.5c, only 11 such elements are formed. The effect of reordering becomes even more significant in large systems where a high percentage of the non-zero elements are created during elimination.

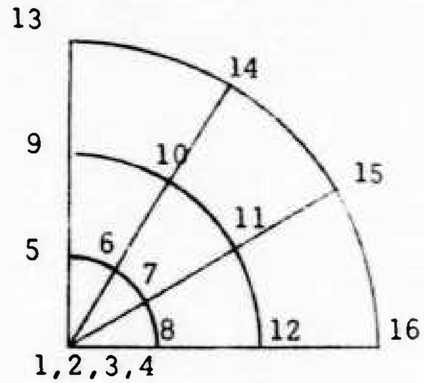
The removal of JLØC from the branch tables of the nodes connected to it (Step 8 of flow chart) is equivalent to the elimination of non-zero elements located directly below the diagonal element in the row of JLØC. Nodes added to the same branch tables (Step 10) on the other hand, represent the non-zero elements created as a result of the activity in Step 8. Steps 8 and 10 make up the pseudo-elimination process which is an essential part of the reordering routine. The final contents of the branch tables represent the off-diagonal non-zero elements of the upper triangular form of the stiffness matrix. The location of these elements has to be defined before the actual elimination can be carried out.

Two arrays are needed to locate a non-zero element. These are NRØW, generated for convenience during the actual elimination process (Figure 1.6),

---

* NREØ in which NREØ(i) gives the original number of the node assigned the new number i.

(a) Finite Element Mesh



Note: x,c = non-zero elements
blank = zero element

(b) Before Gaussian Elimination

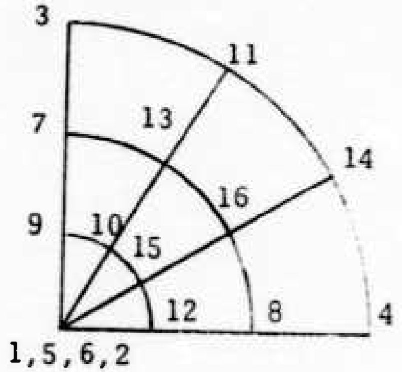
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	x	x			x	x										
2	x	x	x		x	x	x									
3		x	x	x		x	x	x								
4			x	x			x	x								
5	x	x			x	x			x	x						
6	x	x	x		x	x	x		x	x	x					
7		x	x	x		x	x	x		x	x	x				
8			x	x			x	x			x	x				
9					x	x			x	x			x	x		
10					x	x	x		x	x	x		x	x	x	
11						x	x	x		x	x	x		x	x	x
12							x	x			x	x			x	x
13									x	x			x	x		
14									x	x	x		x	x	x	
15										x	x	x		x	x	x
16											x	x			x	x

(c) After Gaussian Elimination

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	x	x			x	x										
2		x	x		x	x	x									
3			x	x	c	x	x	x								
4				x	c	c	x	x								
5					x	x	c	c	x	x						
6						x	x	c	x	x	x					
7							x	x	c	x	x	x				
8								x	c	c	x	x				
9									x	x	c	c	x	x		
10										x	x	c	x	x	x	
11											x	x	c	x	x	x
12												x	c	c	x	x
13													x	x	c	c
14														x	x	c
15															x	x
16																x

Figure 1.4. Unreordered Stiffness Matrix

(a) Finite Element Mesh



Note: x,c = non-zero elements
blank = zero element

(b) Before Gaussian Elimination

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	x				x				x	x						
2		x				x						x			x	
3			x				x				x		x			
4				x				x						x		x
5	x				x	x			x	x					x	
6		x			x	x				x		x			x	
7			x				x		x	x	x		x			
8				x				x				x		x	x	x
9	x				x		x		x	x			x			
10	x				x	x	x		x	x			x		x	x
11			x				x				x		x	x		x
12		x				x		x				x			x	x
13			x				x		x	x	x		x	x	x	x
14				x				x			x		x	x		x
15		x			x	x		x		x		x	x		x	x
16				x				x		x	x	x	x	x	x	x

(c) After Gaussian Elimination

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	x				x				x	x						
2		x				x						x			x	
3			x				x				x		x			
4				x				x						x		x
5					x	x			x	x					x	
6						x			c	x		x			x	
7							x		x	x	x		x			
8								x				x		x	x	x
9									x	x	c	c	x		c	
10										x	c	c	x		x	x
11											x	c	x	x	c	x
12												x	c	c	x	x
13													x	x	x	x
14														x	c	x
15															x	x
16																x

Figure 1.5. Reordered Stiffness Matrix

and LØC, generated in the reordering routine. NRØW(j) gives the number assigned to the first off-diagonal non-zero element in row j. LØC(i) gives the column location of the ith non-zero element. Thus the kth non-zero element would be located in column LØC(k) and in row m where NRØW(m) $\leq$ k < NRØW(m+1). Sequential numbers, starting with one, are used to identify the non-zero elements. The numbering proceeds in the row-wise direction with the off-diagonal elements in a row numbered ahead of the diagonal element. The following tables, used to locate the non-zero elements in Figure 1.5c, serve to clarify the idea presented in this paragraph. The arrows point to the diagonal elements.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LØC(j)	5	9	10	1	6	12	15	2	7	11	13	3	8	14	16	4

j	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
LØC(j)	6	9	10	15	5	9	10	12	15	6	9	10	11	13	7

j	32	33	34	35	36	37	38	39	40	41	42
LØC(j)	12	14	15	16	8	10	11	12	13	15	9

j	43	44	45	46	47	48	49	50	51	52	53	54
LØC(j)	11	12	13	15	16	10	12	13	14	15	16	11

j	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
LØC(j)	13	14	15	16	12	14	15	16	13	15	16	14	16	15	16

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NROW(i)	1	5	9	13	17	22	27	32	37	43	49	55	60	64	67

Table 1.1 Locator Arrays for Non-zero Elements of Figure 1.5c

1.3 Gaussian Elimination

In the discussion in the preceding section, it should be noted that a "non-zero element" is actually a submatrix whose order is equal to the

number of degrees of freedom specified at a node and a "row" are actually rows occupied by the submatrices. Hereafter, these expressions shall convey the same meanings. Thus, a "non-zero element" is either a 2x2 submatrix (2-dimensional cases) or a 3x3 submatrix (3-dimensional cases). Similarly, the "nth row" of the stiffness matrix would be taken to mean the nth 2 rows or the nth 3 rows of the matrix.

The flow chart of the Gaussian elimination routine is shown in Figure 1.6. In Step 3, only the non-zero elements on and to the right of the diagonal need be stored in $[S]$ because of symmetry. The elements are stored in the same order that they would appear in the stiffness matrix with enough spaces reserved for the non-zero elements which will be created later on during the elimination process. Using Figure 1.5c again as an example, $[S]$ would appear as follows during the storage of the different rows of the stiffness matrix:

Row 1:

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} \end{bmatrix}$$

Row 6:

$$\begin{bmatrix} s_{11} & s_{12} & 0 & 0 & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{1,10} \\ s_{21} & s_{22} & 0 & 0 & s_{25} & s_{26} & s_{27} & s_{28} & s_{29} & s_{2,10} \end{bmatrix}$$

Row 9:

$$\begin{bmatrix} s_{11} & s_{12} & 0 & 0 & 0 & 0 & s_{17} & s_{18} & 0 & 0 \\ s_{21} & s_{22} & 0 & 0 & 0 & 0 & s_{27} & s_{28} & 0 & 0 \end{bmatrix}$$

Row 16:

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$$

in which the spaces marked s_{ij} are occupied by the x elements while those marked 0 (zero) are reserved for the c elements.

After the formation of any row, say i, the process of eliminating all non-zero elements in the row located to the left of the diagonal immediately follows. Had the entire upper triangular half of the stiffness matrix been considered in the elimination procedure, these elements would be simply the "mirror image" of the elements lying on the ith column of the preceding rows. Since this is not the case, a method for locating the aforementioned elements has to be devised. This is accomplished by introducing the spotter array NSPT. The spotter array keeps track, for each row, of the element in that row whose "mirror image" is next in line for elimination. Because of the row-wise direction of the elimination procedure, the order in which the elements of any particular row are used in the elimination process proceeds from left to right; that is, the "mirror image" of the left-most element is eliminated first and that of the right-most element is eliminated last. Thus, at the start of the elimination process NSPT(j) = NRØW(j) for any row j and after each elimination the NSPT value of the affected rows is incremented by one. NRØW(j) is defined on page 11. The procedure discussed in this paragraph corresponds to Steps 4 through 6 of the flow chart shown in Figure 1.6.

It should be noted that after elimination of a row, the elements in that row are stored in the next available space in tape or drum #10. Normalization of an element (pre-multiplication of the element by the inverse of the diagonal

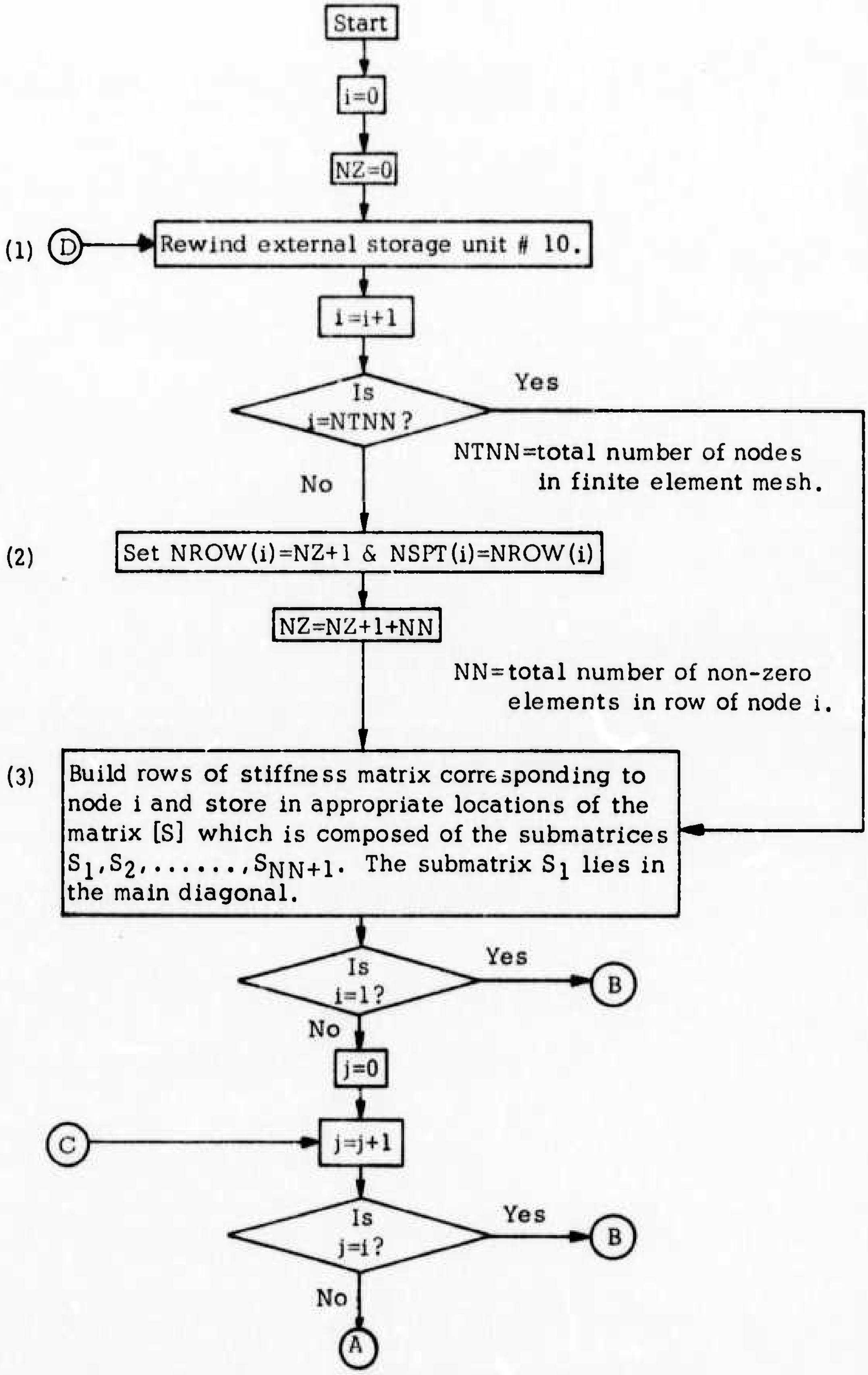


Figure 1.6. Flow Chart of Gaussian Elimination Routine

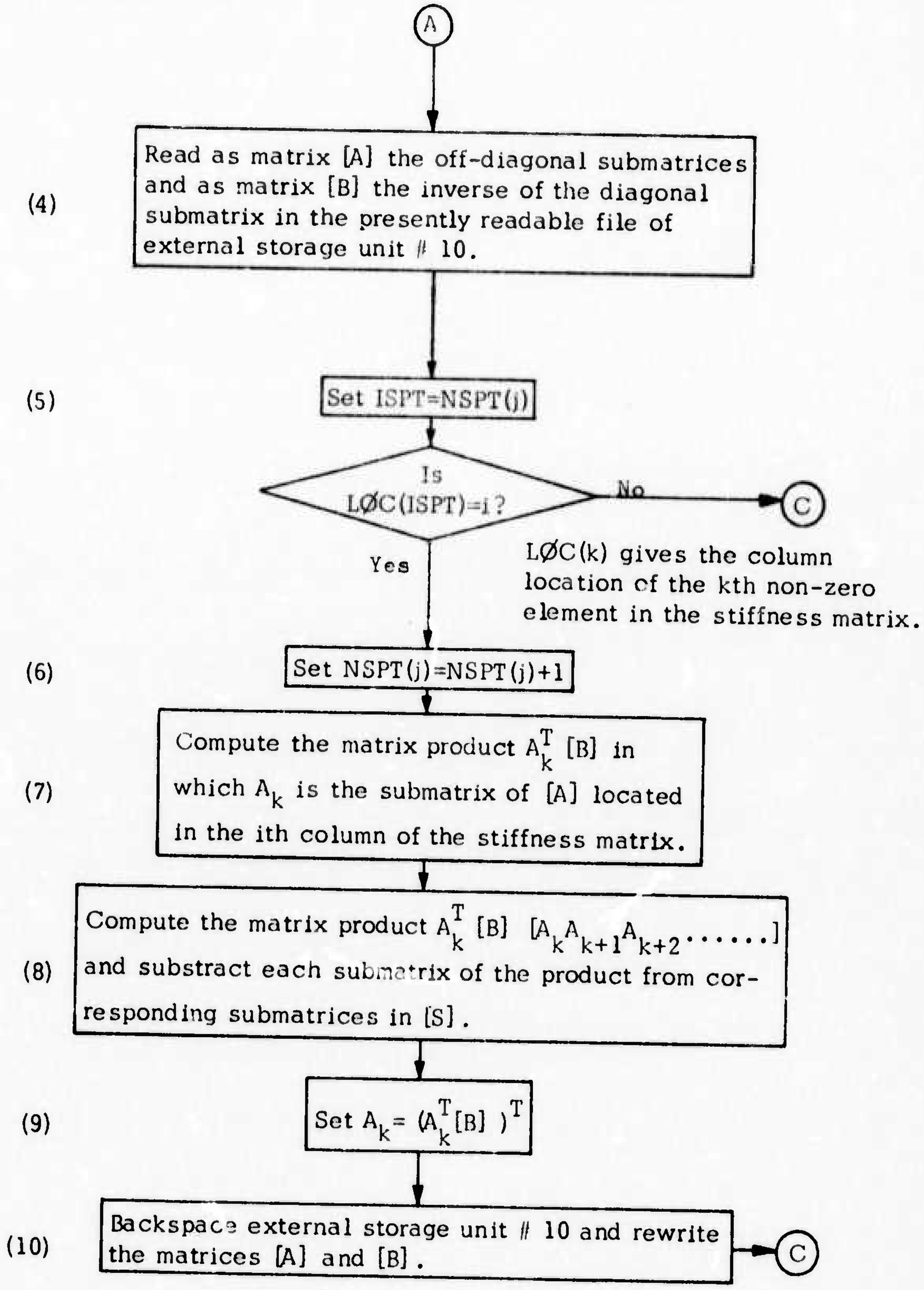


Figure 1.6 Continued.

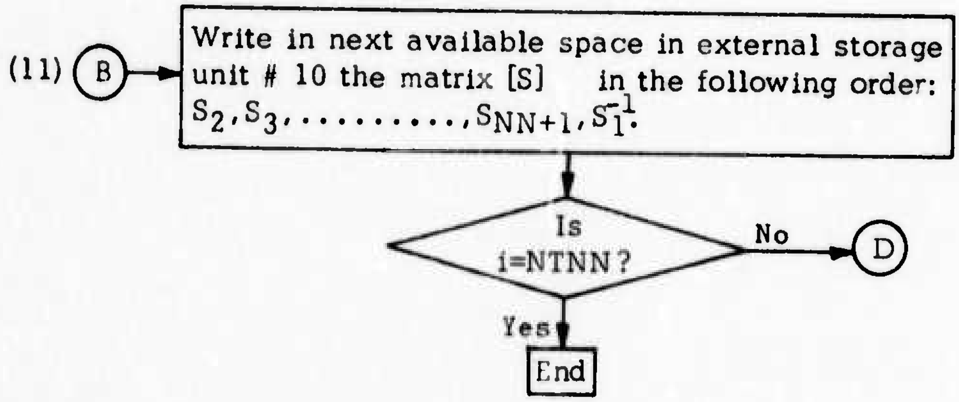


Figure 1.6 Continued.

element) takes place only when the element is no longer needed in the elimination process (Step 9). The actual row operations carried out during the elimination process are represented by Step 8.

2.4 Determination of Principal Stresses in Three Dimensions

It has been determined in Reference 3 that the principal stresses in a 3-dimensional solid are the roots of the cubic equation

$$F(S) = S^3 - a_1 S^2 + a_2 S - a_3 = 0 \tag{1.1}$$

in which

$$a_1 = \sigma_r + \sigma_\theta + \sigma_z$$

$$a_2 = \sigma_r \sigma_\theta + \sigma_\theta \sigma_z + \sigma_n \sigma_z - \tau_{\theta z}^2 - \tau_{r\theta}^2 - \tau_{rz}^2$$

$$a_3 = \sigma_r \sigma_\theta \sigma_z + 2\tau_{\theta z}\tau_{r\theta}\tau_{rz} - \sigma_r \tau_{\theta z}^2 - \sigma_\theta \tau_{rz}^2 - \sigma_z \tau_{r\theta}^2$$

The graph of F(S), with the maxima, minima, and inflection points indicated, is shown in Figure 1.7.

Applying the Newton-Raphson method, the roots of equation (1.1) can be evaluated by means of the iteration formula

$$S_{n+1} = S_n - \frac{F(s_n)}{F'(s_n)} \tag{1.2}$$

in which

$$F'(S) = \frac{dF}{dS} = 3S^2 - 2a_1 S + a_2$$

$n = 0, 1, 2, \ldots\ldots$

S_n = approximation of root value at nth iteration

Setting $S_o = \frac{a_1}{3}$ (that is, starting the iteration at the inflection point I), equation (1.2) will yield the value of the middle root R_2. With R_2 known, equation (1.1) may be rewritten as

$$F(S) = (S-R_2)Q(S) = 0 \tag{1.3}$$

in which Q(S) is a quadratic polynomial whose roots are the two remaining roots of equation (1.1). Q(S) may be conveniently written as

$$Q(S) = S^2 + b_1 S + b_2 \tag{1.4}$$

in which (4)

$$b_1 = R_2 - a_1$$

$$b_2 = R_2 b_1 + a_2$$

Thus the two remaining roots are

$$R_1 = \frac{-b_1 - \sqrt{b_1^2 - 4b_2}}{2}$$

$$R_3 = \frac{-b_1 + \sqrt{b_1^2 - 4b_2}}{2} \tag{1.5}$$

The direction cosines l_i, m_i, n_i of the principal stresses are obtained from the equations

$$(R_i - \sigma_r)l_i - \tau_{r\theta} m_i - \tau_{rz} n_i = 0$$

$$-\tau_{r\theta} l_i + (R_i - \sigma_\theta)m_i - \tau_{\theta z} n_i = 0$$

$$-\tau_{rz} l_i - \tau_{\theta z} m_i + (R_i - \sigma_z) n_i = 0 \tag{1.6}$$

taking into account the relationship

$$l_i^2 + m_i^2 + n_i^2 = 1 \tag{1.7}$$

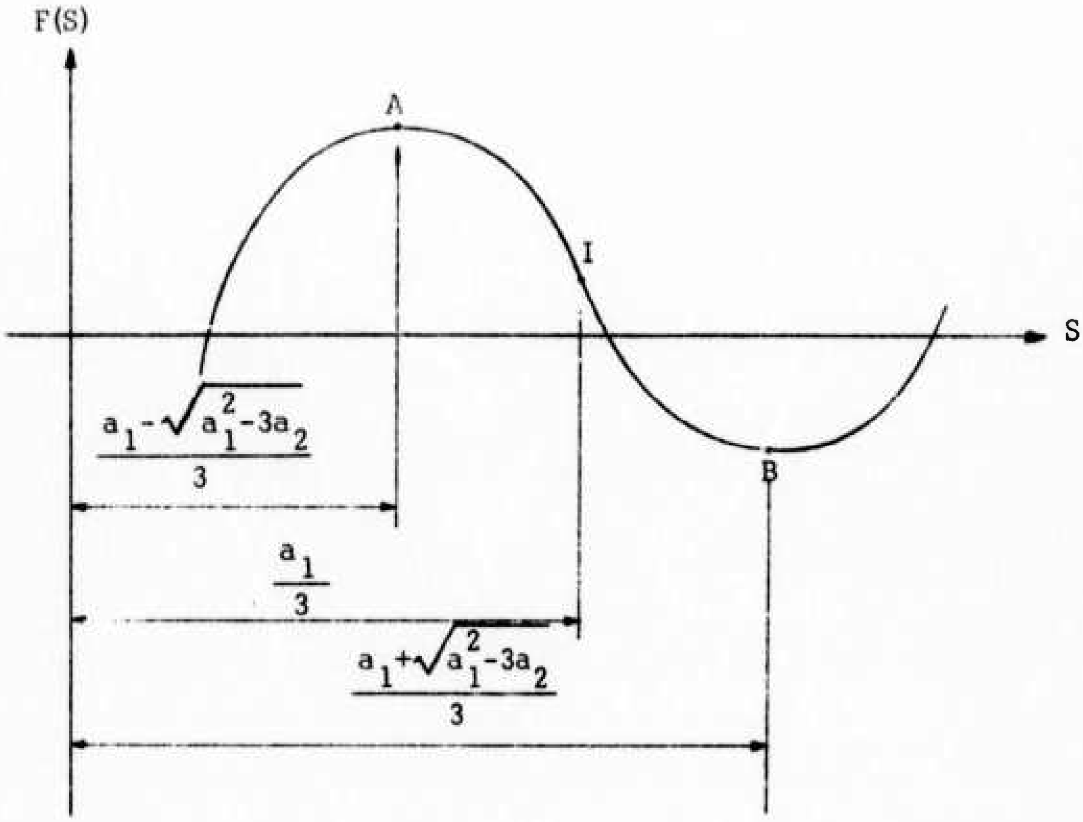


Figure 1.7. Graphical Representation of Cubic Polynomial (Eq. 1.1).

REFERENCES

1. "Aspects of Mechanical Behaviour of Rock Under Static and Cyclic Loading (Part A: Mechanical Behaviour of Rock Under Static Loading)", Semi-Annual Technical Progress Report, Engineering Experiment Station, University of Wisconsin, Madison, Wisconsin, September 1972.

2. Jensen, H.G. and G.A. Parks, "Efficient Solutions for Linear Matrix Equations", Journal of the Structural Division, Proc. ASCE, Vol. 96, No. ST1, Jan. 1970, pp. 49-64.

3. Timoshenko, S. and J.N. Goodier, Theory of Elasticity, McGraw-Hill Book Company, Inc., New York, 1951, p. 217.

4. Henrici, P., Elements of Numerical Analysis, John Wiley & Sons, New York, 1964, p. 86.

CHAPTER 2

NUMERICAL EXAMPLES

2.1 Introduction

To demonstrate the capabilities of the program, several cylinders exhibiting all three rock properties considered in the program are analyzed. Unfortunately, it is not possible to compare the results, particularly with respect to the three-dimensional, anisotropic, and non-homogeneous cases, with any existing alternative solution, so relatively coarse meshes are used to save on computer expense which quite prohibitive in most cases. Isotropic problems are run mainly to compare the load-displacement and load-strain curves obtained by the two failure criteria used in the program. In the new failure criterion, the stiffness matrix of the elements that failed is revised by reducing to zero the elastic modulus across tension cracks when failure is in tension or by removing a fixed percentage (specified by program user) of said matrix when the failure is in compression or shear. In the old failure criterion (used in semi-annual report), the stiffness matrix of the failed elements is reduced by a fixed percentage regardless of whether tensile, compressive or shearing failure occurs. One shortcoming of the old criterion is the unrealistic shape of the resulting load-strain curves and, to a certain extent, load-displacement curves.

In anisotropic discs and cylinders the allowable stresses at a point vary with direction making the determination of the load factor (critical ratio of actual stress to allowable stress) of an element quite involved, one requiring time-consuming trial-and-error procedure. Because of this, a simplifying assumption is made in the program in which the critical direction is considered to coincide with one of the principal stresses as in the case of isotropic discs and cylinders. This assumption makes it necessary only to

determine the allowable stress in the direction of the critical principal stress to obtain the element load factor. In the program the allowable stresses at a point are assumed to vary as an ellipsoidal surface with the principal axes coinciding with the directions of anistropy.

Considerable effort was expended to check each individual subroutine of the program by long hand calculation. As part of the debugging process, it was also decided to rerun the isotropic problem presented as Problem 2 in the semi-annual report. The results were in close agreement with those obtained by the previous program.

All the cylinders treated herein possess the following dimensions:

Diameter = 3 inches

Length = 1 inch

"Vertical displacement" refers to the average displacement of the loaded nodes in the direction of the load while "horizontal strain" refers to the strain at the central point of the disc or cylinder measured normal to the loaded plane.

2.2 Two-dimensional Anisotropic Problem

Elastic Moduli:

E_1 = 3,000,000 psi

E_2 = 5,700,000 psi

Allowable Compressive Stresses:

C_1 = 15,000 psi

C_2 = 27,000 psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratio = .25

Shear Modulus = 1,600,000. psi

Finite Element Mesh (Whole Disc):

5 radial divisions
16 circumferential divisions

Directions of anisotropy are indicated in Figure 2.1.

2.3 Two-dimension Isotropic Problem (Old Failure Criterion)

Elastic Modulus = 5,700,000. psi

Allow. Compressive Stress = 27,000. psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratio = .25

Finite Element Mesh (Disc Quadrant):

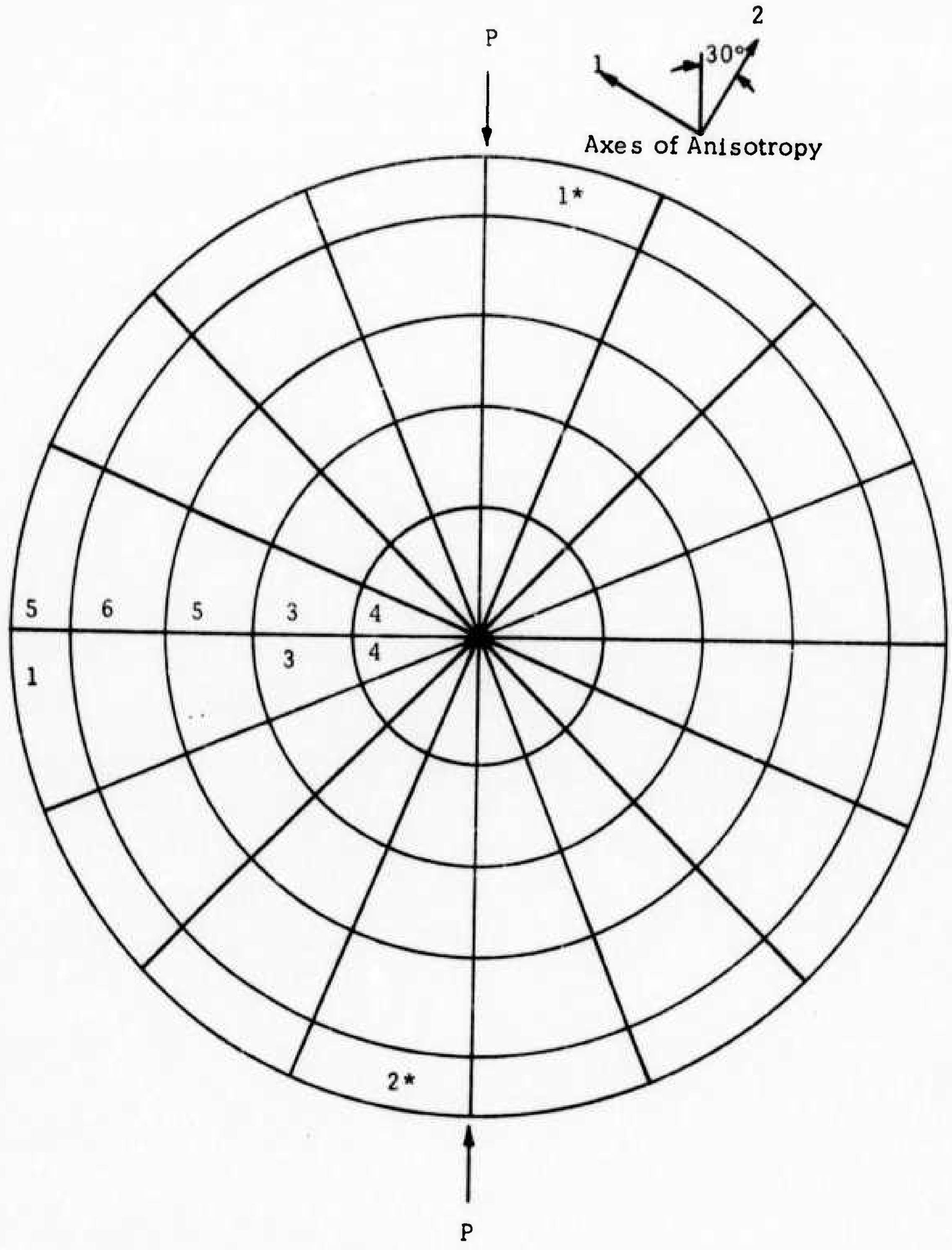
6 radial divisions
9 circumferential divisions

Stiffness matrices of failed elements are reduced 90%

INPUT DATA LISTING FOR PROBLEM 2.2

Column

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 55	56 60	61 65	66 70	71	Card No.
1	5	16	0	1	2	6	1	0							1
.05		.10													2
.90		.05		3.		1.		1							3
5	21	5	21												4
0.		.4		.7		1.		1.3		1.5					5a
0.		22.5		45.		67.5		90.		112.5		135.		157.5	6a
180.		202.5		225.		247.5		270.		292.5		315.		337.5	6b
30.		0.		0.		15000.		27000.		27000.		3000000.		5700000.	9
5700000.		.25		.23		.23		1600000.		2800000.		2800000.			10



* Failed in shear.
The rest failed in tension.

Figure 2.1 Progression of Failure in Problem 2.2

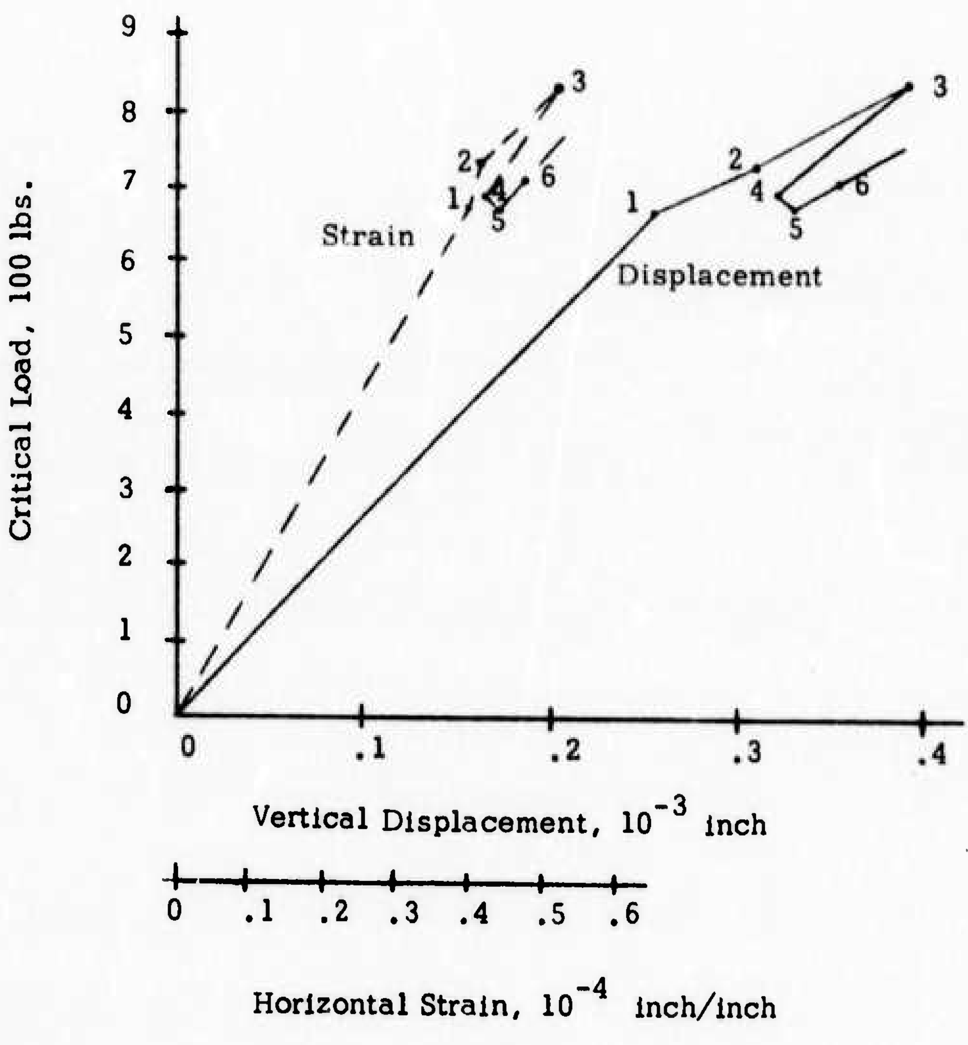
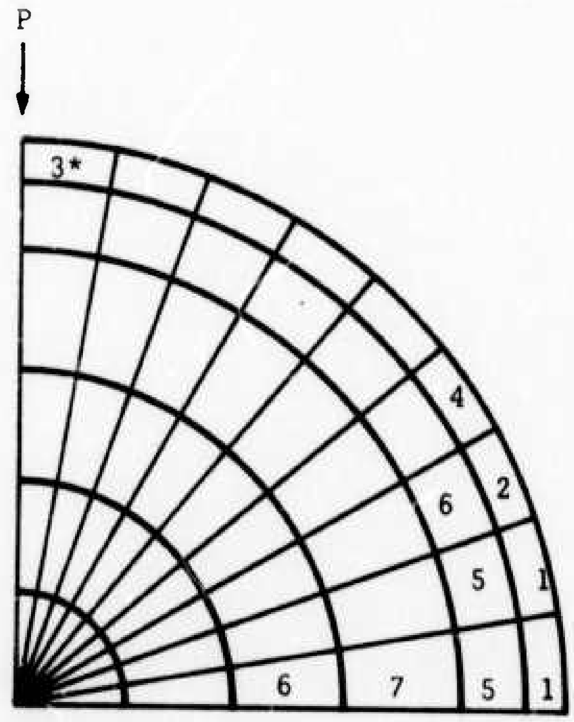


Figure 2.2 Load-Displacement-Strain Curves (Problem 2.2)

INPUT DATA LISTING FOR PROBLEM 2.3

Column

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 60	61 70	71	Card No.
-1	6	9	0	0	2	7	9	0					1
.05		.10		5700000.		5700000.		27000.		27000.	.25	.25	2
.90		.05		3.		1.		1					3
10	20	10	20										4
0.		.3		.6		.9		1.2		1.4	1.5		5a
0.		10.		20.		30.		40.		50.	60.	70.	6a
80.		90.											6b



*Failed in shear. Rest failed in tension.

Figure 2.3. Progression of Failure in Problem 2.3.

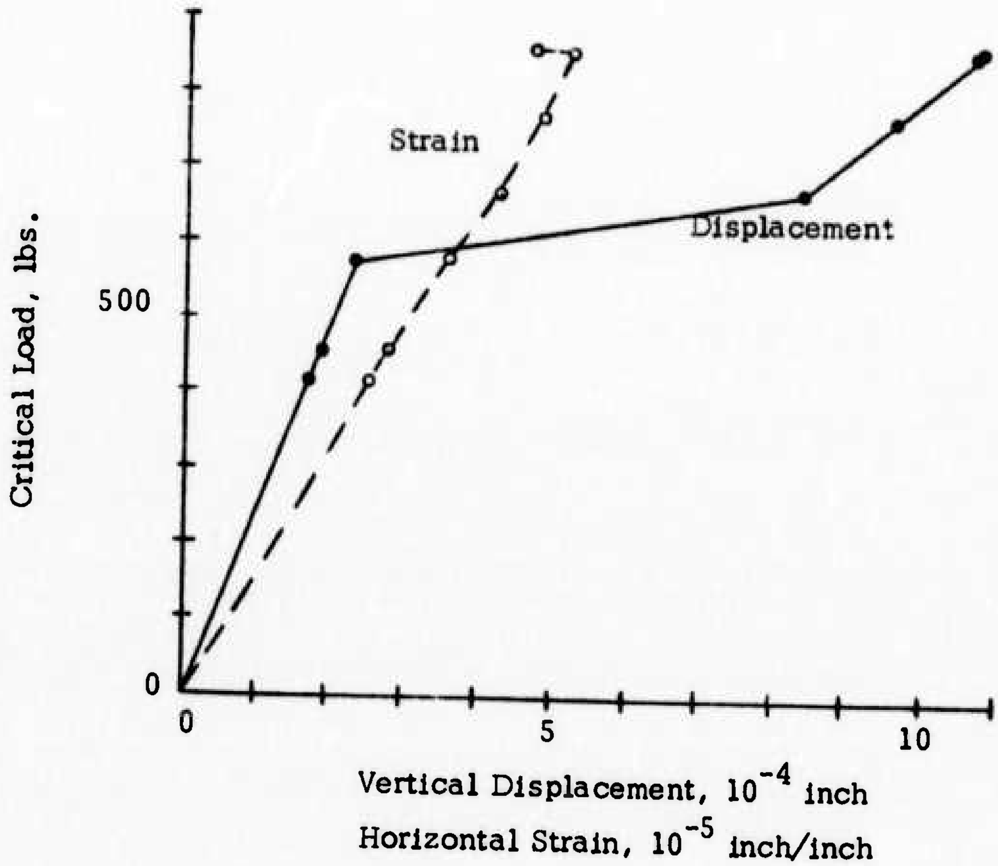


Figure 2.4. Load-Displacement-Strain Curves (Problem 2.3)

2.4 Two-dimensional Isotropic Problem (New Failure Criterion)

Data are the same as Problem 2.3. Stiffness matrices of elements failing in compression or shear are reduced 90%. Elastic moduli across tension cracks are reduced to zero.

2.5 Three-dimensional Isotropic Problem (Old Failure Criterion)

Elastic Modulus = 5,700,000. psi

Allow. Compressive Stress = 27,000. psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = 9.00 of Allow. Compression

Poisson's Ratio = .25

Finite Element Mesh (Cylinder Quadrant):

5 radial divisions
6 circumferential divisions
2 axial (Z-axis) divisions

Because of symmetry only half the length of cylinder is considered in analysis. Stiffness matrices of failed elements are reduced 90%.

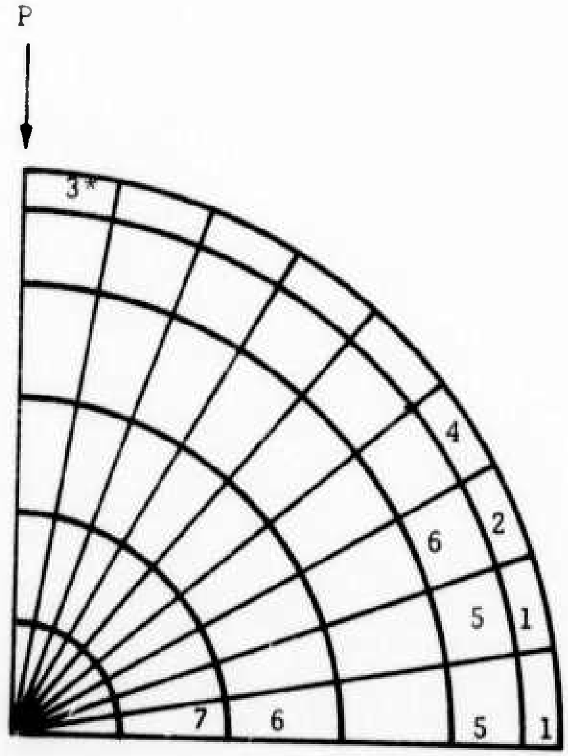
2.6 Three-dimensional Isotropic Problem (New Failure Criterion)

Data are the same as Problem 2.5. Stiffness matrices of elements failing in compression or shear are reduced 90%. Elastic moduli across tension cracks are reduced to zero.

INPUT DATA LISTING FOR PROBLEM 2.4

Column

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 60	61 70	71	Card No.
-1	6	0	0	0	2	7	9	0					1
.05		.10		5700000.		5700000.		27000.		27000.	.25	.25	2
.90		.05		3.		1.		0					3
10	20	10	20										4
0.		.3		.6		.9		1.2		1.4	1.5		5a
0.		10.		20.		30.		40.		50.	60.	70.	6a
80.		90.											6b



*Failed in shear. Rest failed in tension.

Figure 2.5. Progression of Failure in Problem 2.4.

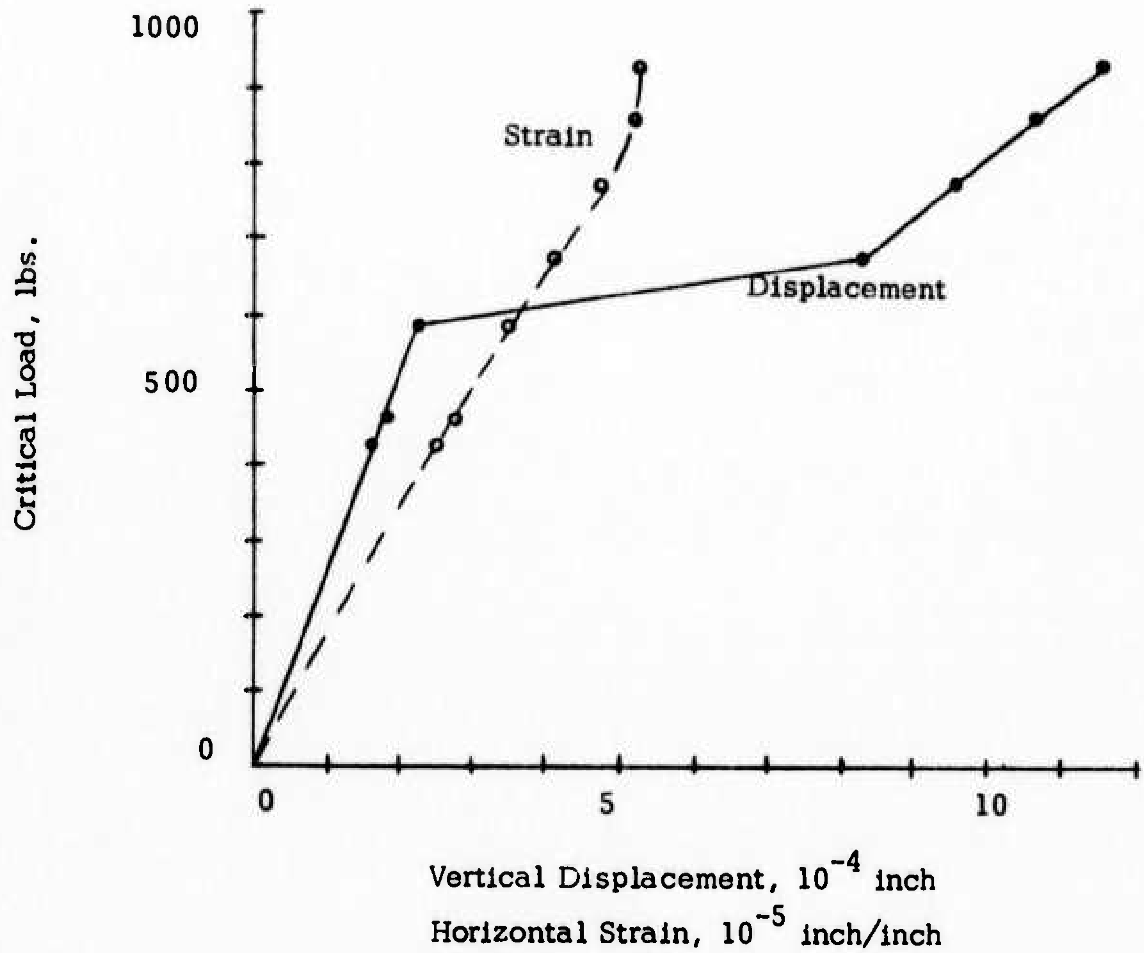
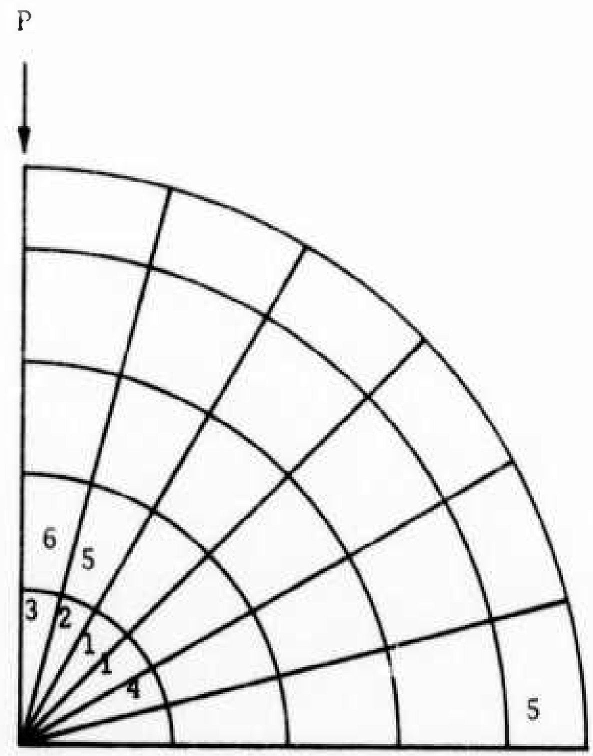


Figure 2.6. Load-Displacement-Strain Curves (Problem 2.4)

INPUT DATA LISTING FOR PROBLEM 2.5

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 60	61 70	71	Card No.
-1	5	6	1	0	3	6	6	0					1
.05		9.		5700000.		5700000.		27000.		27000.	.25	.25	2
.90		.05		3.		1.		1					3
7	21	14	28										4
0.		.4		.7		1.		1.3		1.5			5a
0.		15.		30.		45.		60.		75.	90.		6a
0.		.5											7a



Note: All elements failed in tension.

Figure 2.7. Progression of Failure in Problem 2.5.

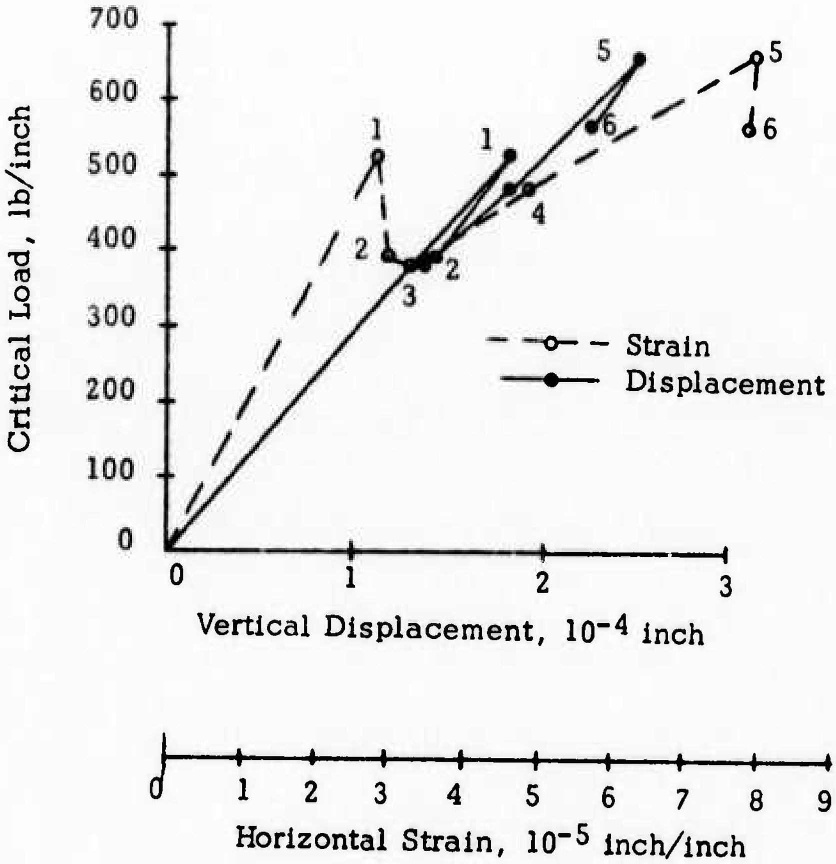
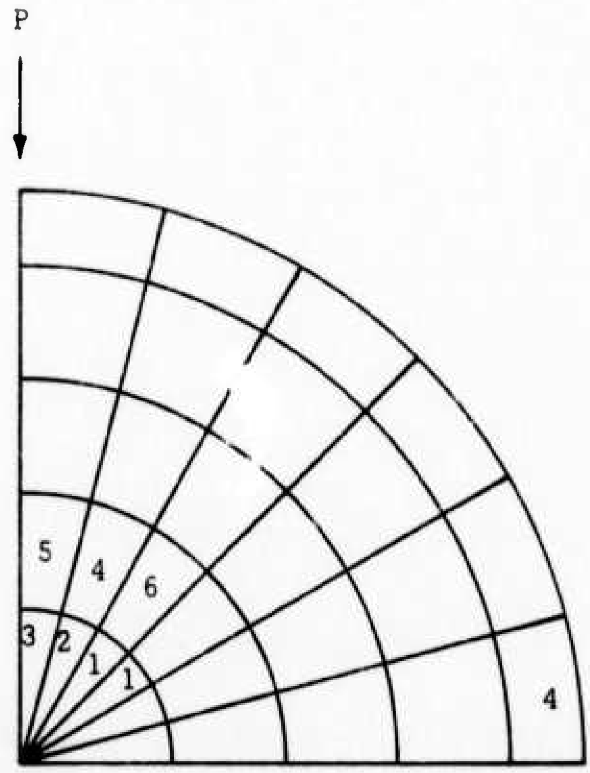


Figure 2.8. Load-Displacement-Strain Curves (Problem 2.5).

INPUT DATA LISTING FOR PROBLEM 2.6

Column

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 60	61 70	71	Card No.
-1	5	6	1	0	3	6	6	0					1
.05		9.		5700000.		5700000.		27000.		27000.	.25	.25	2
.90		.05		2.		1.							3
7	21	14	28										4
0.		.4		.7		1.		1.2		1.5			5a
0.		15.		30.		45.		60.		75.	90.		6a
0.		.5											7a



Note: All elements failed in tension

Figure 2.9. Progression of Failure in Problem 2.6.

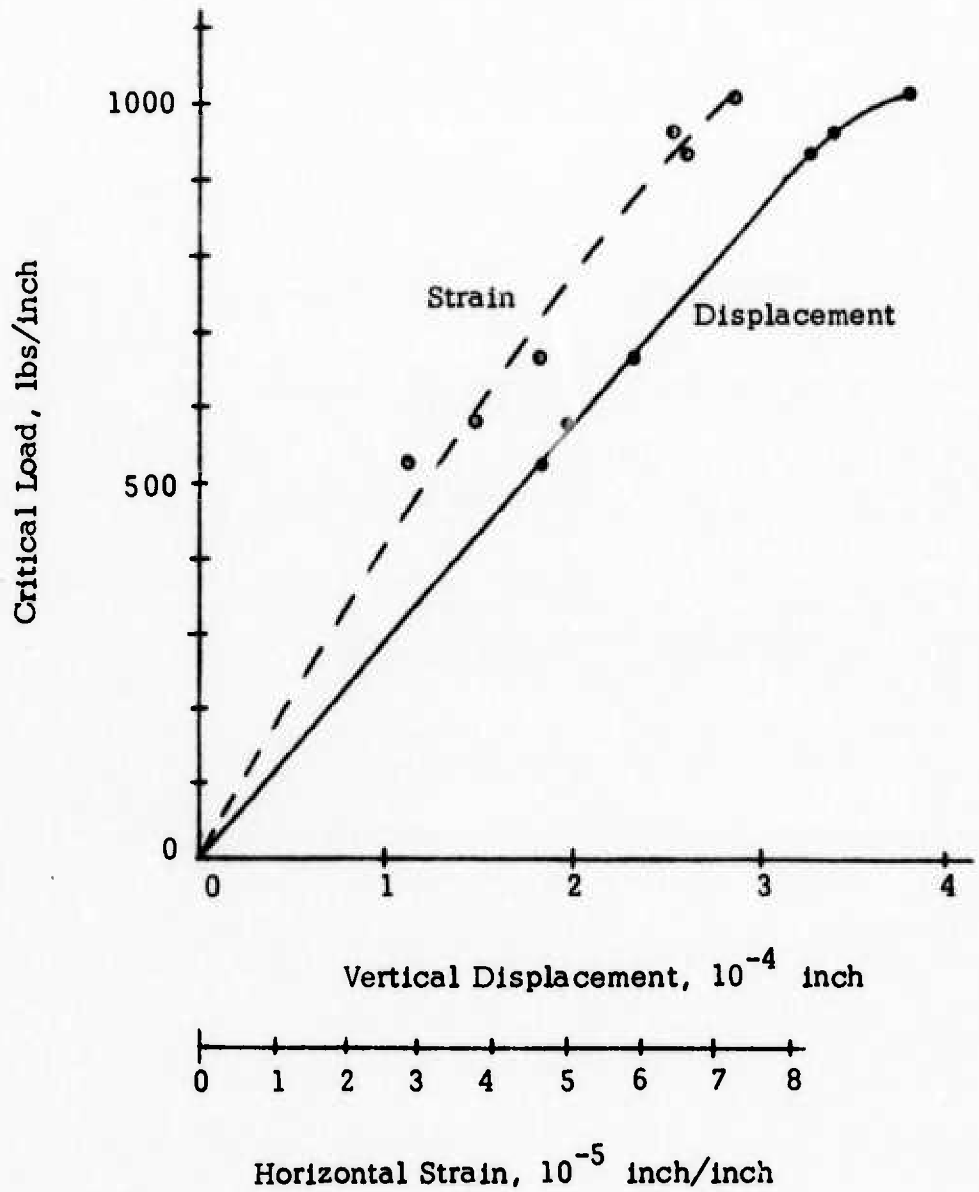


Figure 2.10. Load-Displacement-Strain Curves (Problem 2.6).

2.7 Three-dimensional Anisotropic Problem

Elastic Moduli:

E_1 = 3,000,000. psi

E_2 = 5,700,000. psi

E_3 = 5,700,000. psi

Allow. Compressive Stresses:

C_1 = 15,000. psi

C_2 = 27,000. psi

C_3 = 27,000. psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratios:

μ_1 = .25

μ_2 = .23

μ_3 = .23

Shear Moduli:

G_1 = 1,600,000 psi

G_2 = 2,800,000 psi

G_3 = 2,800,000 psi

Finite Element Mesh (Whole Cylinder):

3 radial divisions
8 circumferential divisions
2 axial (Z-axis) divisions

Directions of anisotropy are indicated in Figure 2.11.

INPUT DATA LISTING FOR PROBLEM 2.7

Column

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 60	61 70	71	Card No.
1	3	8	2	1	3	5	1	0					1
.05		.10											2
.90		.05		3.		1.		1	1				3
3	27	19	43										4
0.		.6		1.1		1.5							5a
0.		45.		90.		135.		180.		225.	270.	315.	6a
0.		.5		1.									7a
30.		10.		0.		15000.		27000.		27000.	3000000.	5700000.	9
5700000.		.25		.23		.23		1600000.		2800000.	2800000.		10

Table 2.1. Nodal Reordering for Problem 2.7.

RESULTS OF NODAL REORDERING

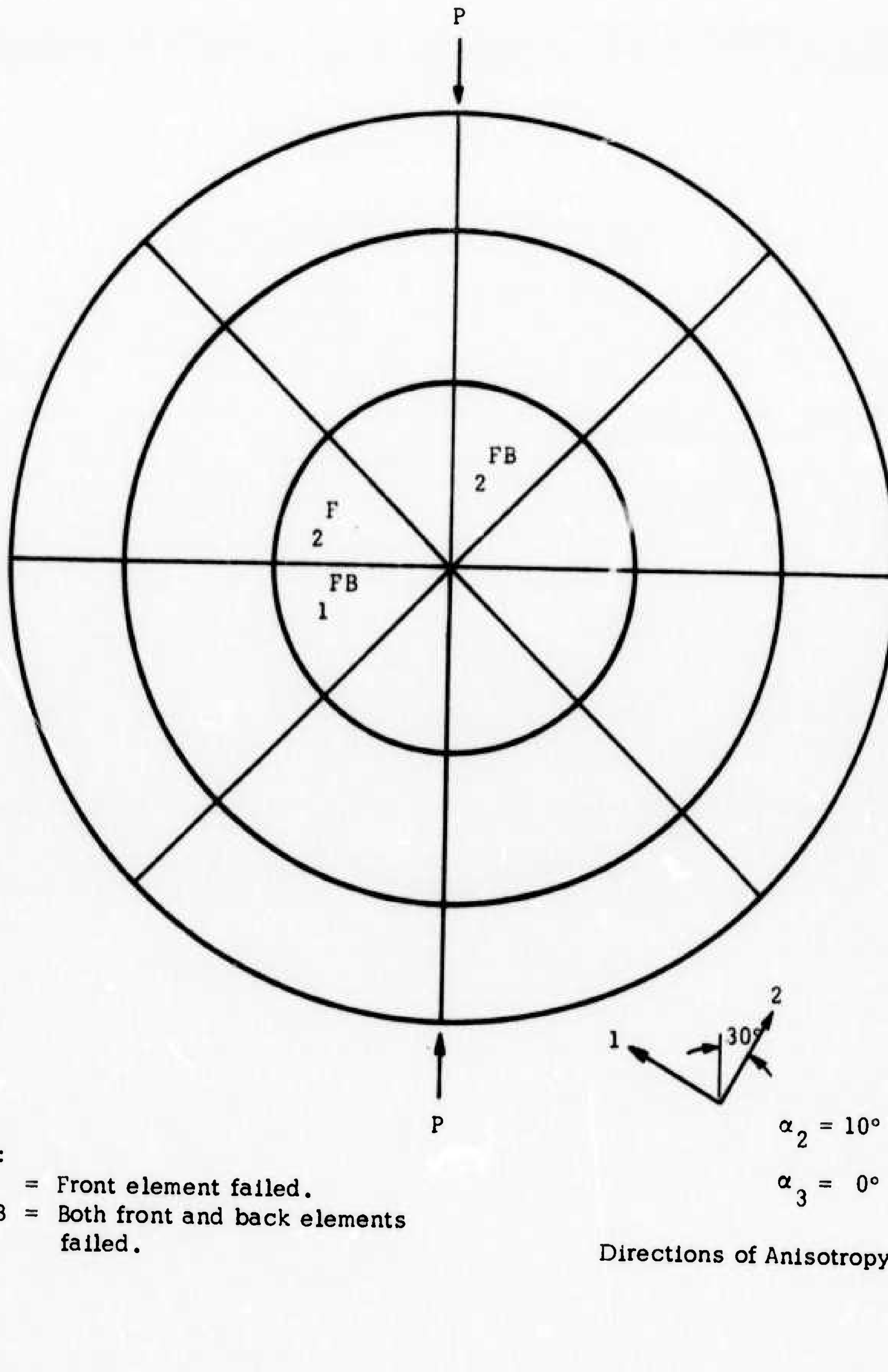
NEW NO.	ORIG. NO.	NODAL CONNECTIONS (ORIG. NOS.)
1	1	9 25 33 2 26 10 34 16 40 8 32
2	3	11 27 35 2 26 10 34 4 36 12 28
3	5	13 29 37 4 36 12 28 30 6 14 38
4	7	15 31 39 16 40 8 32 30 6 14 38
5	17	9 33 41 18 10 42 34 24 48 16 40
6	19	11 35 43 18 10 42 34 44 20 36 12
7	21	13 37 45 44 20 36 12 22 14 46 38
8	23	15 39 47 24 48 16 40 22 14 46 38
9	73	81 49 57 74 82 50 58 80 88 64 56
10	75	83 51 59 74 82 50 58 76 84 60 52
11	77	85 53 61 76 84 60 52 78 86 62 54
12	79	87 55 63 80 88 64 56 78 86 62 54
13	89	81 65 57 82 90 66 58 88 96 72 64
14	91	83 67 59 82 90 66 58 84 92 68 60
15	93	85 69 61 84 92 68 60 86 94 70 62
16	95	87 71 63 88 96 72 64 86 94 70 62
17	9	25 33 41 2 18 26 10 42 34 24 48 16 40 8 32
18	11	27 35 43 2 18 26 10 42 34 4 44 20 36 12 28
19	13	29 37 45 4 44 20 36 12 28 22 30 6 14 46 38
20	15	31 39 47 24 48 16 40 8 32 22 30 6 14 46 38
21	81	65 49 57 74 82 90 66 50 58 80 88 96 72 64 56
22	83	67 51 59 74 82 90 66 50 58 76 84 92 68 60 52
23	85	69 53 61 76 84 92 68 60 52 78 86 94 70 62 54
24	87	71 55 63 80 88 96 72 64 56 78 86 94 70 62 54
25	25	33 41 49 57 2 18 26 10 50 42 34 58 64 56 24 48 16 40 8 32
26	27	35 43 51 59 2 18 26 10 50 42 34 58 4 60 52 44 20 36 12 28
27	29	37 45 53 61 4 60 52 44 20 36 12 28 22 30 6 14 62 46 54 38
28	31	39 47 55 63 64 56 24 48 16 40 8 32 22 30 6 14 62 46 54 38
29	65	33 41 49 57 74 82 90 66 50 42 34 58 80 88 96 72 64 56 48 40
30	67	35 43 51 59 74 82 90 66 50 42 34 58 76 84 92 68 60 52 44 36
31	69	37 45 53 61 76 84 92 68 60 52 44 36 78 86 94 70 62 46 54 38
32	71	39 47 55 63 80 88 96 72 64 56 48 40 78 86 94 70 62 46 54 38
33	33	41 49 57 2 74 18 82 26 90 10 66 50 42 34 58 80 88 96 72 64 56 24 48 16 40 8 32
34	41	49 57 2 74 18 82 26 90 10 66 50 42 34 58 80 88 96 72 64 56 24 48 16 40 8 32
35	49	57 2 74 18 82 26 90 10 66 50 42 34 58 80 88 96 72 64 56 24 48 16 40 8 32
36	57	2 74 18 82 26 90 10 66 50 42 34 58 80 88 96 72 64 56 24 48 16 40 8 32
37	35	43 51 59 2 74 18 82 26 90 10 66 50 42 34 58 4 76 84 92 68 60 52 44 20 36 12 28
38	43	51 59 2 74 18 82 26 90 10 66 50 42 34 58 4 76 84 92 68 60 52 44 20 36 12 28
39	51	59 2 74 18 82 26 90 10 66 50 42 34 58 4 76 84 92 68 60 52 44 20 36 12 28
40	59	2 74 18 82 26 90 10 66 50 42 34 58 4 76 84 92 68 60 52 44 20 36 12 28
41	37	45 53 61 4 76 84 92 68 60 52 44 20 36 12 28 78 22 86 30 94 6 70 14 62 46 54 38
42	45	53 61 4 76 84 92 68 60 52 44 20 36 12 28 78 22 86 30 94 6 70 14 62 46 54 38
43	53	61 4 76 84 92 68 60 52 44 20 36 12 28 78 22 86 30 94 6 70 14 62 46 54 38

Table 2.1 Continued.

44	61	4 54	76 38	84	92	68	60	52	44	20	36	12	28	78	22	86	30	94	6	70	14	62	46
45	39	47 14	55 62	63 46	80 54	88 38	96	72	64	56	24	48	16	40	8	32	78	22	86	30	94	6	70
46	47	55 62	63 46	80 54	88 38	96	72	64	56	24	48	16	40	8	32	78	22	86	30	94	6	70	14
47	55	63 46	80 54	88 38	96	72	64	56	24	48	16	40	8	32	78	22	86	30	94	6	70	14	62
48	63	80 54	88 38	96	72	64	56	24	48	16	40	8	32	78	22	86	30	94	6	70	14	62	46
49	2	74 56	18 52	82 24	26 44	90 48	10 20	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60
50	74	18 52	82 24	26 44	90 48	10 20	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56
51	18	82 24	26 44	90 48	10 20	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52
52	82	26 44	90 48	10 20	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24
53	26	90 48	10 20	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44
54	90	10 20	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48
55	10	66 16	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48	20
56	66	50 36	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48	20	16
57	50	42 40	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48	20	16	36
58	42	34 12	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48	20	16	36	40
59	34	58 8	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48	20	16	36	40	12
60	58	4 28	80 32	76	88	84	96	92	72	68	64	60	56	52	24	44	48	20	16	36	40	12	8
61	4	80 32	76 78	88 22	84 86	96 30	92 94	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28
62	80	76 78	88 22	84 86	96 30	92 94	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32
63	76	88 22	84 86	96 30	92 94	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78
64	88	84 86	96 30	92 94	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22
65	84	96 30	92 94	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86
66	96	92 94	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86	30
67	92	72 6	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86	30	94
68	72	68 70	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86	30	94	6
69	68	64 14	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86	30	94	6	70
70	64	60 62	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86	30	94	6	70	14
71	60	56 46	52 54	24 38	44	48	20	16	36	40	12	8	28	32	78	22	86	30	94	6	70	14	62
72	56	52	24	44	48	20	16	36	40	12	8	28	32	78	22	86	30	94	6	70	14	62	46

Table 2.1 Continued.

```
                 54  38
 73      52      24  44  48  20  16  36  40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54
                 38
 74      24      44  48  20  16  36  40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38

 75      44      48  20  16  36  40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 76      48      20  16  36  40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 77      20      16  36  40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 78      16      36  40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 79      36      40  12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 80      40      12   8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 81      12       8  28  32  78  22  86  30  94   6  70  14  62  46  54  38
 82       8      28  32  78  22  86  30  94   6  70  14  62  46  54  38
 83      28      32  78  22  86  30  94   6  70  14  62  46  54  38
 84      32      78  22  86  30  94   6  70  14  62  46  54  38
 85      78      22  86  30  94   6  70  14  62  46  54  38
 86      22      86  30  94   6  70  14  62  46  54  38
 87      86      30  94   6  70  14  62  46  54  38
 88      30      94   6  70  14  62  46  54  38
 89      94       6  70  14  62  46  54  38
 90       6      70  14  62  46  54  38
 91      70      14  62  46  54  38
 92      14      62  46  54  38
 93      62      46  54  38
 94      46      54  38
 95      54      38
 96      38

PERCENTAGE OF NON-ZERO TERMS = 21.09

EXTERNALLY-LOADED NODES --  73,  77,  89,  93,  81,  85,

BOUNDARY NODES --  73,  77,  89,  93,  81,  85,

NODE    1 (ORIG. NO.    1)
   ELEMENTS CONNECTED TO NODE --    1,   8,   0,   0,   0,   0,   0,   0,
   NODES OF ELEMENT    1 --    1,   2,  26,  25,   9,  10,  34,  33,
   NODES OF ELEMENT    8 --    8,   1,  25,  32,  16,   9,  33,  40,

NODE    2 (ORIG. NO.    3)
   ELEMENTS CONNECTED TO NODE --    3,   2,   0,   0,   0,   0,   0,   0,
   NODES OF ELEMENT    3 --    3,   4,  28,  27,  11,  12,  36,  35,
   NODES OF ELEMENT    2 --    2,   3,  27,  26,  10,  11,  35,  34,

NODE    3 (ORIG. NO.    5)
   ELEMENTS CONNECTED TO NODE --    5,   4,   0,   0,   0,   0,   0,   0,
   NODES OF ELEMENT    5 --    5,   6,  30,  29,  13,  14,  38,  37,
   NODES OF ELEMENT    4 --    4,   5,  29,  28,  12,  13,  37,  36,

NODE    4 (ORIG. NO.    7)
   ELEMENTS CONNECTED TO NODE --    7,   6,   0,   0,   0,   0,   0,   0,
   NODES OF ELEMENT    7 --    7,   8,  32,  31,  15,  16,  40,  39,
   NODES OF ELEMENT    6 --    6,   7,  31,  30,  14,  15,  39,  38,

NODE    5 (ORIG. NO.   17)
   ELEMENTS CONNECTED TO NODE --    0,   0,  16,   9,   0,   0,   0,   0,
   NODES OF ELEMENT   16 --   16,   9,  33,  40,  24,  17,  41,  48,
```



Note:
F = Front element failed.
FB = Both front and back elements failed.

Figure 2.11. Progression of Failure in Problem 2.7.

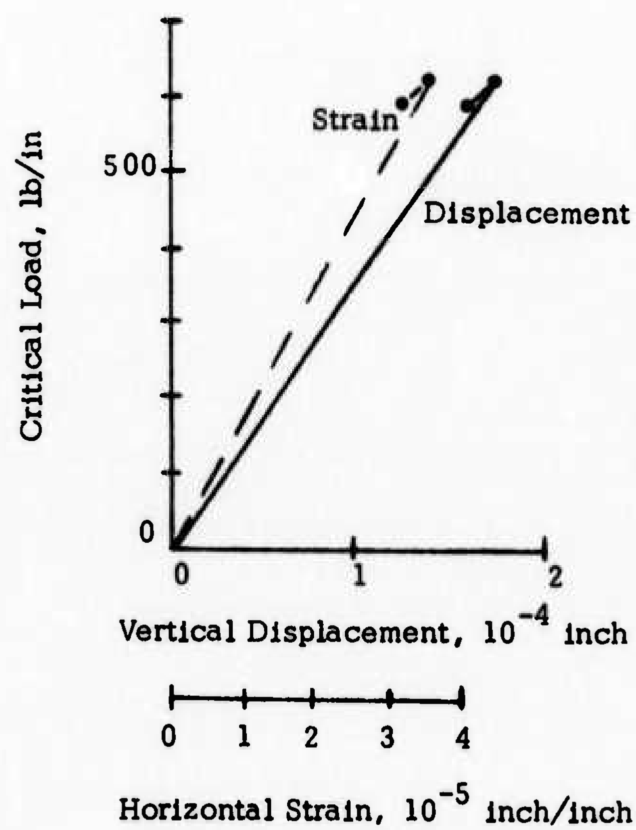


Figure 2.12. Load-Displacement-Strain Curves (Problem 2.7)

2.8 Three-dimensional Non-homogeneous Problem

Elastic Modulus = 5,500,000 to 6,500,000 psi

Allow. Compressive Stress = 22,000 to 32,000 psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratio = .23 to .25

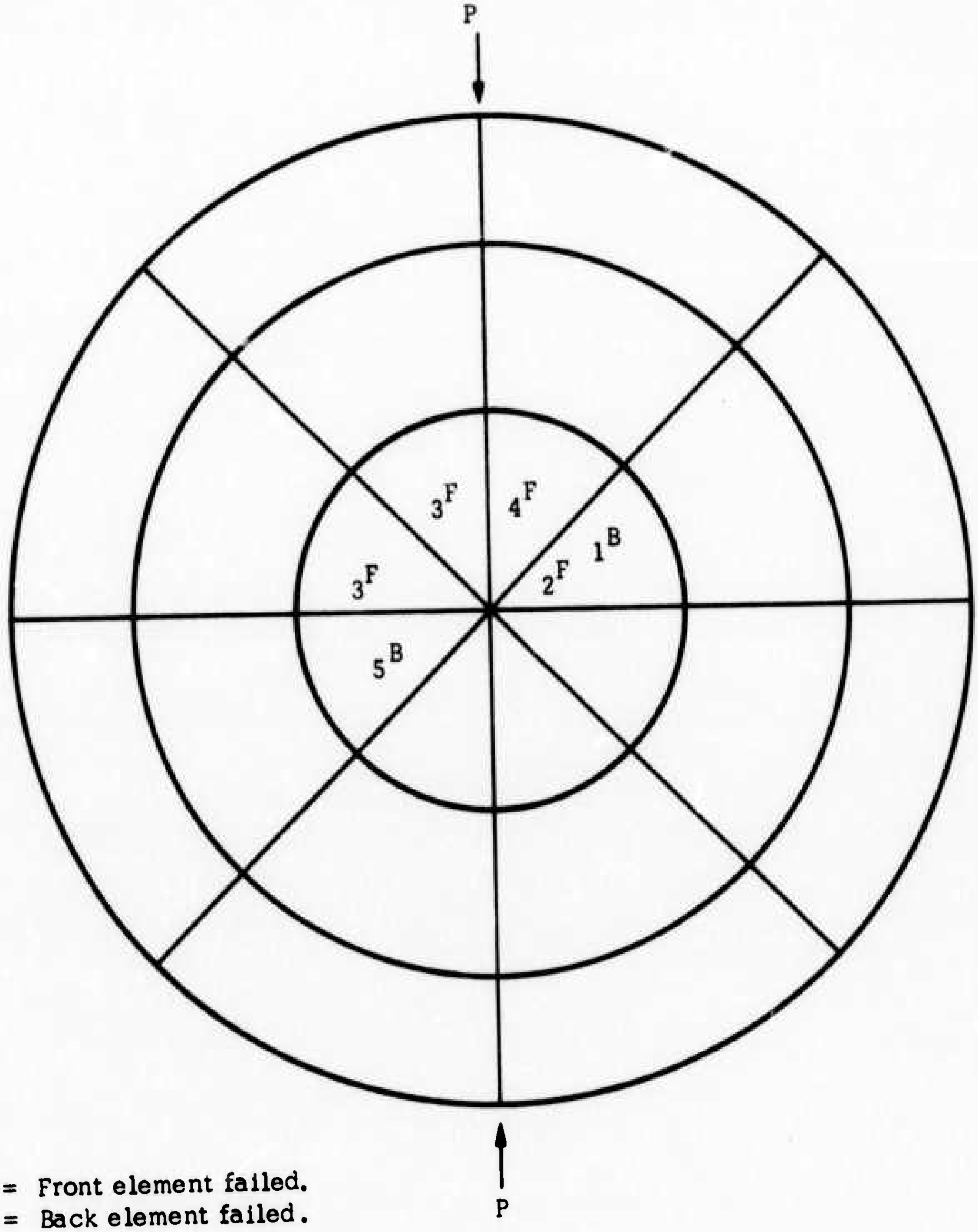
Finite Element Mesh (Whole Cylinder):

3 radial divisions
8 circumferential divisions
2 axial divisions

INPUT DATA LISTING FOR PROBLEM 2.8

Column

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50	51 60	61 70	71	Card No.
0	3	8	2	1	3	6	1	0					1
.05		.10		5500000.		6500000.		22000.		32000.	.23	.25	2
.90		.05		3.		1.		0	1				3
3	27	19	43										4
0.		.6		1.1		1.5							5a
0.		45.		90.		135.		180.		225.	270.	315.	6a
0.		.5		1.									7a
39554	31329												8



Note:
F = Front element failed.
B = Back element failed.

Figure 2.13. Progression of Failure in Problem 2.8.

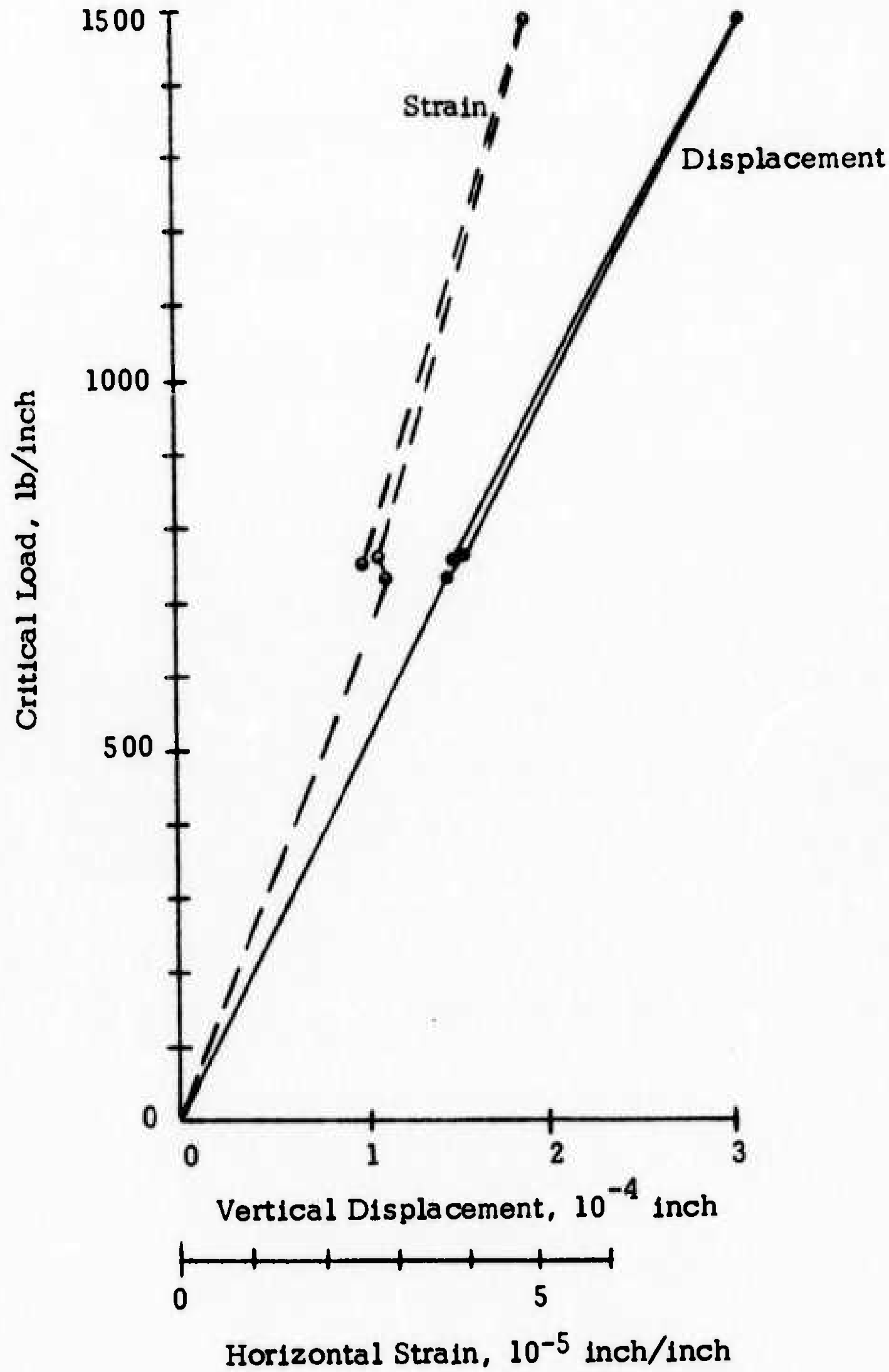


Figure 2.14. Load-Displacement-Strain Curves (Problem 2.8)

2.9 Discussion of Results and Conclusion

It is difficult to verify explicitly the theoretical correctness of the present solution due to lack of an alternative solution to compare the results with. To compensate for this situation, extensive long-hand checks were undertaken on the individual routines that make up the program. The results, on the whole, are fairly reasonable and seem to indicate the correctness of the whole program and the theory upon which it is based.

The unrealistic failure shown in Fig. 2.1 could be attributed to the following:

1) Coarseness of the finite element mesh.
2) The simplifying assumption used in anisotropic cases (second paragraph of section 2.1) that the critical direction is necessarily along one of the principal stresses.

Item 1) above could also be the cause of the unrealistic failure patterns shown in Figs. 2.3 and 2.5. It should be noted that although the meshes shown in these figures appear to be relatively fine, they are actually approximately six times coarser than the mesh of Problems two and three of the semi-annual report (see page 49 of same report). Unrealistic failure pattern were also observed during the debugging of the previous two-dimensional program when coarse meshes were used.

Problems 2.3 through 2.6 were run to determine the relative merits of the two failure criteria used in the program. There is no significant difference between the two as far as the failure patterns are concerned. The displacement and strain curves, however, differ considerably particularly in the coarser three-dimensional mesh (see Figures 2.8 and 2.10). The new failure criterion seems to yield consistently definable curves while the old criterion does not. Thus although the failure patterns are unrealistic, the plots show a much better relationship between load and strain and load and displacement.

Table 2.1 shows the results of the nodal reordering carried out in connection with Problem 2.7. It took no more than 30 seconds to complete the whole reordering process.

Table 2.2 gives some information concerning the sizes of the test problems described in this chapter. The maximum number of nodes and non-zero elements of the stiffness matrix that the program can handle are indicated in Table 2.3. These limits may still be increased depending on the computer used.

All mathematical operations in the program are carried out in single precision. It may be necessary to resort to double precision to reduce rounding-off errors.

Table 2.2 Test Problems Comparison

Problem	Number of Nodes	Dimension	Execution Time (min.)	Number of Load Cycles	% of Non-zero Elements
2.2	96	2	3.10	6	12.27
2.3	70	2	1.55	7	11.20
2.4	70	2	1.55	7	11.20
2.5	84	3	5.60	6	15.56
2.6	84	3	5.91	6	15.56
2.7	96	3	9.00	2	21.09
2.8	96	3	15.00	5	21.09

Table 2.3 Program Limits

	2-Dim	3-Dim
No. of Nodes	500	250
No. of Finite Elements	480	480
No. of Non-zero Elements in Stiffness Matrix	25,000	25,000

APPENDIX A
PROGRAM LISTING

The program listed in the following pages is written in Fortran V language for use specifically in the UNIVAC 1108 Computer located at the University of Wisconsin - Madison campus. It requires six external storage units (either tape or drum) which are assigned the numbers 10, 11, 12, 13, 14, and 15. The contents of the files written in units 10 and 12 continually undergo changes during a program run making it necessary for these particular units to posses random access capabilities.

In addition to the subroutines listed, two other subroutines, RANUN(R) and RANUNS(N), are called in the program during execution of non-homogeneous problems. RANUN(R) generates pseudo-random numbers assuming a statistically uniform distribution while RANUNS(N) sets to N the starting point of the random number generator; N is supplied by the user. Both subroutines were developed by the Madison Academic Computing Center (MACC) and are part of the MACC Random Number Routine package.

Except for the replacement of RANUN(R) and RANUNS(N), there should be no difficulty in making this program adaptable to other computer systems.

```
C     MAIN PROGRAM
C     ******************************************
      COMMON/FUNKY/NBRC(500),NREO(500),NOER(500)
      COMMON/RUTH/LOC(25000),NTNN,NB,NX,MINUM
      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
      COMMON/TANAI/RO,RS,TO,TS,ZS,NY,HNB
      COMMON/LEYTE/RC(30),TC(50),ZC(30)
      COMMON/BGAND/C(8,8),B(6,24),H
      COMMON/SANNA/PD(480,3),FMAT(3,8)
      COMMON/ABALN/D(3,600)
      COMMON/NANAY/CR(3,3),NW
      COMMON/JANET/NSPD,NWSD(100),SPD(100,3)
      COMMON/BORAP/NLOD(100),CNLD(100),NE,NER
      COMMON/TATAY/ISO,ALF1,ALF2,ALF3,E1,E2,E3,P1,P2,P3,
     1              G1,G2,G3
      DIMENSION BRC(500),CRT(3,3),CSN(3,3),LA(20),
     1LAC(25000),NM(3),NOD(8),P(3),PT(3),S(3,300),
     2SL(24,24),SLD(24,24),STRS(6),VV(3),WW(3)
      EQUIVALENCE (D(1,301),S),(LOC,LAC)
      DATA VV/'COMPRE','TENSIO','SHEARI'/
      DATA WW/'SSION','N','NG'/
      DATA (C(1,I),I=1,8)/1.,1.,1.,1.,1.,1.,1.,1./
      DATA (C(2,I),I=1,8)/-1.,-1.,1.,1.,-1.,-1.,1.,1./
      DATA (C(3,I),I=1,8)/-1.,1.,1.,-1.,-1.,1.,1.,-1./
      DATA (C(4,I),I=1,8)/1.,-1.,1.,-1.,1.,-1.,1.,-1./
      DATA (C(5,I),I=1,8)/-1.,-1.,-1.,-1.,1.,1.,1.,1./
      DATA (C(6,I),I=1,8)/1.,1.,-1.,-1.,-1.,-1.,1.,1./
      DATA (C(7,I),I=1,8)/1.,-1.,-1.,1.,-1.,1.,1.,-1./
      DATA (C(8,I),I=1,8)/-1.,1.,-1.,1.,1.,-1.,1.,-1./
C     READ IN DATA
C     ISO = -1 , HOMOGENEOUS AND ISOTROPIC
C     ISO =  0 , NONHOMOGENEOUS
C     ISO =  1 , HOMOGENEOUS AND ANISOTROPIC
C     NWOP = 0 , PORTION OF DISC IS ANALYZED
C     NWOP = 1 , WHOLE DISC IS ANALYZED
C     NZ=0 FOR 2-DIMENSIONAL PROBLEMS
      READ 100,ISO,NR,NS,NZ,NWOP,NDIM,NCYC,NER,NSS
      READ 102,TCRT,SCRT,EM1,EM2,CS1,CS2,PR1,PR2
      READ 104,POSM,TRN,DIA,H,NTEP,NGOT
      READ 100,NST1,NST2,MST1,MST2
      LRM=ISO
      NIDA=NDIM-1
      INGA=(NDIM-1)*50
      INGO=500/(NDIM-1)
      NR1=NR+1
      NT=NS-NWOP
      NS1=NT+1
      NZ1=NZ+1
      READ 102,(RC(I),I=1,NR1)
      READ 102,(TC(I),I=1,NS1)
      IF (NDIM .EQ. 2)GO TO 3
      READ 102,(ZC(I),I=1,NZ1)
    3 PRINT 300
      NTNN=(NT+1)*(NZ+1)*(NR+1)
      NELEM=NR*NS
```

```
      IF (NDIM .EQ. 3)NELEM=NELEM*NZ
      IF (ISO)4,6,8
    4 DO 5 I=1,NELEM
      PD(I,1)=EM1
      PD(I,2)=PR1
    5 PD(I,3)=CS1
      PRINT 301,EM1
      PRINT 302,PR1
      PRINT 303,CS1
      GO TO 9
    6 READ 106,N
      CALL RANUNS(N)
      DO 7 I=1,NELEM
      CALL UNIFRM(EM1,EM2,EMS)
      CALL UNIFRM(PR1,PR2,PRS)
      CALL UNIFRM(CS1,CS2,CSS)
      PD(I,1)=EMS
      PD(I,2)=PRS
    7 PD(I,3)=CSS
      PRINT 306,EM1,EM2
      PRINT 307,PR1,PR2
      PRINT 308,CS1,CS2
      GO TO 9
    8 READ 102,ALF1,ALF2,ALF3,C1,C2,C3,E1,E2
      READ 102,E3,P1,P2,P3,G1,G2,G3
      PRINT 326,ALF1,ALF2,ALF3
      PALO=C1*C2
      PAL=PALO
      ALF1=ALF1*.0174532925
      ALF2=ALF2*.0174532925
      ALF3=ALF3*.0174532925
      CS2=COS(ALF2)
      SN2=SIN(ALF2)
      CS3=COS(ALF3)
      SN3=SIN(ALF3)
      IF (NDIM .EQ. 2)GO TO 2
      PALO=PALO*C3
      CRT(3,1)=-SN2
      CRT(3,2)=CS2*SN3
      CRT(3,3)=CS2*CS3
    2 PRINT 309,E1,E2,E3
      PRINT 310,P1,P2,P3
      PRINT 311,G1,G2,G3
      PRINT 312,C1,C2,C3
    9 NB=4*(NDIM-1)
      NW=0
      HNB=NB
      NX=NB*NDIM
      NY=(NDIM-1)*3
      DO 10 I=1,NY
      DO 10 J=1,NX
   10 B(I,J)=0.
      DO 14 I=1,3
      DO 14 J=1,8
   14 FMAT(I,J)=0.
      PRINT 304,TCRT
      PRINT 305,SCRT
```

```
      PRINT 313
      PRINT 314,DIA
      PRINT 315,H
      PRINT 319,NELEM
      PRINT 320,NTNN
      PRINT 316,(RC(I),I=1,NR1)
      PRINT 317,(TC(I),I=1,NS1)
      DO 15 I=2,NS1
   15 TC(I)=TC(I)*.0174532925
      IF (NDIM .EQ. 2)GO TO 16
      PRINT 318,(ZC(I),I=1,NZ1)
   16 CALL NDTND(LAC,INGA,INGO)
      CALL REORD(LAC,INGA,INGO)
      IF (MINUM .GT. INGA)GO TO 433
C     GENERATE EXTERNALLY-LOADED NODES (NLOD) AND
C          MAGNITUDES OF LOADS (CNLD)
      IF (NSS .EQ. 0)NSS=NS/2
      NE=NZ+1
      ND=(NT+1)*NE*NR
      CLLD=-500.
      IF (NWOP .EQ. 1)CLLD=-1000.
      CRES=0.
      NF=-NT
      IF (NZ .EQ. 0)GO TO 19
      DO 18 I=1,NZ
      NF=NF+NT+1
      NLOD(I)=NF+ND
      CDUM=.5*(ZC(I+1)-ZC(I))*CLLD
      CNLD(I)=CDUM+CRES
      CRES=CDUM
   18 CONTINUE
      CNLD(NE)=CRES
   19 IF (NZ .EQ. 0)CNLD(1)=CLLD
      NLOD(NE)=ND+NF+NT+1
      IF (NWOP .NE. 1)GO TO 22
      DO 20 I=1,NE
      NLOD(I+NE)=NLOD(I)+NSS
   20 CNLD(I+NE)=CNLD(I)
      NE=NE*2
   22 NLOD(NE+1)=0
      NE1=NE-1
      IF (NE1 .EQ. 0)GO TO 25
      DO 24 I=1,NE1
      NLI=NLOD(I)
      I1=I+1
      DO 24 J=I1,NE
      NLJ=NLOD(J)
      IF (NOER(NLJ) .GT. NOER(NLI))GO TO 24
      T1=NLOD(I)
      T2=CNLD(I)
      NLOD(I)=NLOD(J)
      CNLD(I)=CNLD(J)
      NLOD(J)=T1
      CNLD(J)=T2
      NLI=NLOD(I)
   24 CONTINUE
   25 PRINT 200,(NLOD(I),I=1,NE)
```

```
      NAY=(NT+1)*(NZ+1)
C     GENERATE BOUNDARY NODES (NWSD) AND MAGNITUDES
C          OF SPECIFIED DISPLACEMENTS (SPD)
      IF (NWOP .EQ. 1)GO TO 30
      MD=(NT+1)*(NZ+1)
      DO 26 I=1,MD
      NWSD(I)=I
   26 SPD(I,1)=0.
      MR=NR*(NZ+1)
      MF=MD+MR
      NDR=MD-NT
      DO 27 I=1,MR
      NWSD(MD+I)=NDR+NT+1
      NDR=NWSD(MD+I)
      NWSD(MF+I)=NDR+NS
      SPD(MD+I,1)=200.
   27 SPD(MF+I,1)=200.
      NSPD=MF+MR
      DO 28 I=1,NSPD
      SPD(I,2)=0.
   28 SPD(I,3)=200.
      IF (NGOT .EQ. 1 .OR. NDIM .EQ. 2)GO TO 85
      DO 81 J=1,NSPD
      I=NWSD(J)
      J1=(I-1)/NAY
      J2=NAY*J1+1
      J3=(I-J2)/(NT+1)
      IF (J3 .EQ. 0)SPD(J,3)=0.
   81 CONTINUE
      DO 83 I=1,NTNN
      J1=(I-1)/NAY
      J2=NAY*J1+1
      J3=(I-J2)/(NT+1)
      J4=I-J2-J3*(NT+1)
      IF (J3 .NE. 0)GO TO 83
      IF (J4 .EQ. 0 .OR. J4 .EQ. NT)GO TO 83
      IF (J1 .EQ. 0)GO TO 83
      NSPD=NSPD+1
      NWSD(NSPD)=I
      SPD(NSPD,1)=200.
      SPD(NSPD,2)=200.
      SPD(NSPD,3)=0.
   83 CONTINUE
      GO TO 85
   30 NSPD=NE
      DO 32 J=1,NSPD
      NWSD(J)=NLOD(J)
      SPD(J,1)=200.
      SPD(J,2)=0.
   32 SPD(J,3)=200.
      IF (NGOT .EQ. 1 .OR. NDIM .EQ. 2)GO TO 34
      DO 82 J=1,NSPD
      I=NWSD(J)
      J1=(I-1)/NAY
      J2=NAY*J1+1
      J3=(I-J2)/(NT+1)
      IF (J3 .EQ. 0)SPD(J,3)=0.
```

```
   82 CONTINUE
      DO 84 I=1,NTNN
      J1=(I-1)/NAY
      J2=NAY*J1+1
      J3=(I-J2)/(NT+1)
      J4=I-J2-J3*(NT+1)
      IF (J3 .NE. 0)GO TO 84
      IF (J1 .EQ. NR .AND. J4 .EQ. 0)GO TO 84
      IF (J1 .EQ. NR .AND. J4 .EQ. NSS)GO TO 84
      NSPD=NSPD+1
      NWSD(NSPD)=I
      SPD(NSPD,1)=200.
      SPD(NSPD,2)=200.
      SPD(NSPD,3)=0.
   84 CONTINUE
   85 NS1=NSPD-1
      DO 29 I=1,NS1
      NWI=NWSD(I)
      I1=I+1
      DO 29 J=I1,NSPD
      NWJ=NWSD(J)
      IF (NOER(NWJ) .GT. NOER(NWI))GO TO 29
      T1=NWSD(I)
      T2=SPD(I,1)
      T3=SPD(I,2)
      T4=SPD(I,3)
      NWSD(I)=NWSD(J)
      SPD(I,1)=SPD(J,1)
      SPD(I,2)=SPD(J,2)
      SPD(I,3)=SPD(J,3)
      NWSD(J)=T1
      SPD(J,1)=T2
      SPD(J,2)=T3
      SPD(J,3)=T4
      NWI=NWSD(I)
   29 CONTINUE
   34 PRINT 201,(NWSD(I),I=1,NSPD)
C     FORM STIFFNESS MATRICES OF TYPICAL ELEMENTS AND
C          STORE IN AUXILIARY UNITS 13,14,15
      IF (NER .EQ. 1)GO TO 40
      MINUM=NELEM/NER/3
      IF (NELEM .GT. MINUM*NER*3)MINUM=MINUM+1
      MINAM=MINUM
      JILL=0
      DO 38 I=1,NELEM,NER
      JILL=JILL+1
      IDRUM=(JILL-1)/MINUM+13
      CALL NODES(NDIM,I,NOD)
      NDI=NOD(1)
      NDF=NOD(NB-1)
      CALL INTEG(NDIM,I,NDI,NDF,SL)
      WRITE (IDRUM)((SL(L1,L2),L1=1,NX),L2=1,NX)
      PRINT 400,I,((SL(L1,L2),L1=1,8),L2=1,8)
   38 CONTINUE
  400 FORMAT (25H0STIFF. MATRIX OF ELEMENT,I4/(8E11.5))
C     BUILD UP GLOBAL STIFFNESS MATRIX AND SOLVE FOR
C          NODAL DISPLACEMENTS
```

```
   40 CALL FACTOR
      DO 41 I=1,NELEM
   41 BRC(I)=1.
      DO 80 MI=1,NCYC
      REWIND 12
      MX=0
      PRINT 202,MI
      CRLF=0.
      MINUM=NTNN
      DO 43 I=1,NELEM
      IF (BRC(I) .LE. 0.)GO TO 43
      CALL NODES(NDIM,I,NOD)
      NDI=NOD(1)
      NDF=NOD(NB-1)
      CALL COORD3(NDI,RDI,TTI,ZZI)
      CALL COORD3(NDF,RDF,TTF,ZZF)
      RO=.5*(RDF+RDI)
      RS=.5*(RDF-RDI)
      ZS=.5*(ZZF-ZZI)
      IF (TTF .LT. TTI)TTF=6.283185307
      TO=.5*(TTF+TTI)
      TS=.5*(TTF-TTI)
      CALL STRESS(I,NOD,STRS)
      CALL PSTRES(NDIM,LRM,STRS,P)
      IF (LRM .NE. 1)GO TO 31
      AI=ALFI-TO
      CRT(1,1)=CS2*SIN(AI)
      CRT(2,1)=-CS2*COS(AI)
      CRT(1,2)=CS3*COS(AI)+SN3*SN2*SIN(AI)
      CRT(2,2)=CS3*SIN(AI)-SN3*SN2*COS(AI)
      IF (NDIM .EQ. 2)GO TO 11
      CRT(1,3)=CS3*SN2*SIN(AI)-SN3*COS(AI)
      CRT(2,3)=-CS3*SN2*COS(AI)-SN3*SIN(AI)
   11 DO 23 IA=1,NDIM,NIDA
      DO 23 IB=1,NDIM
      DUM=0.
      DO 21 K=1,NDIM
   21 DUM=DUM+CR(K,IA)*CRT(K,IB)
      CSN(IA,IB)=DUM*DUM
   23 CONTINUE
      POL1=CSN(NDIM,1)*C2*C2+CSN(NDIM,2)*C1*C1
      POL2=CSN(1,1)*C2*C2+CSN(1,2)*C1*C1
      IF (NDIM .EQ. 2)GO TO 12
      POL1=POL1*C3*C3+CSN(NDIM,3)*PAL*PAL
      POL2=POL2*C3*C3+CSN(1,3)*PAL*PAL
   12 PD(I,3)=PALO/SQRT(POL2)
      PD(I,1)=PALO/SQRT(POL1)
      CCCT=PD(I,1)*TCRT
      CCCS=SCRT*(PD(I,3)+PD(I,1))
      GO TO 33
   31 CCCT=PD(I,3)*TCRT
      CCCS=2.*PD(I,3)*SCRT
   33 RT1=-P(1)/PD(I,3)
      RT2=P(NDIM)/CCCT
      RT3=(P(NDIM)-P(1))/CCCS
      BRC(I)=AMAX1(RT1,RT2,RT3)
      IF (BRC(I) .GT. CRLF)CRLF=BRC(I)
```

```
   43 CONTINUE
C     DETERMINE ELEMENTS THAT FAILED, LA(I)
      II=0
      DO 44 I=1,NELEM
      IF (BRC(I) .LE. 0.)GO TO 44
      IF (BRC(I)/CRLF .LT. (1.-TRN))GO TO 44
      II=II+1
      LA(II)=I
      BRC(I)=-1.
   44 CONTINUE
C     CRITICAL LOAD
      CRL=1000./CRLF
      PRINT 199,CRL
      N1=NTNN-NOER(NST1)+1
      N2=NTNN-NOER(NST2)+1
      M1=NTNN-NOER(MST1)+1
      M2=NTNN-NOER(MST2)+1
      TEMP=(RC(2)-RC(1))*CRLF
      STRN=(D(1,N2)-D(1,N1))/TEMP
      STRM=(D(1,M2)-D(1,M1))/TEMP
      AVE=.5*(STRN+STRM)
      PRINT 203
      PRINT 204,STRN
      PRINT 205,STRM
      PRINT 206,AVE
      REWIND 13
      REWIND 14
      REWIND 15
      NM(1)=0
      NM(2)=0
      NM(3)=0
      PRINT 207
      NLF=(NE+1)/2
      DO 42 I=1,NLF
      NLO=NLOD(I)
      NUN=NTNN-NOER(NLO)+1
      CNL1=CNLD(I)*CRL/CLLD
      DZ1=D(1,NUN)*CRL/1000.
      IF (NE .EQ. 1)GO TO 45
      J=I+NLF
      IF (J .GT. NE)GO TO 45
      NLA=NLOD(J)
      NAN=NTNN-NOER(NLA)+1
      CNL2=CNLD(J)*CRL/CLLD
      DZ2=D(1,NAN)*CRL/1000.
      PRINT 209,NLO,CNL1,DZ1,NLA,CNL2,DZ2
      GO TO 42
   45 PRINT 209,NLO,CNL1,DZ1
   42 CONTINUE
      PRINT 211
      IF (NDIM .EQ. 2)PRINT 214
      IF (NDIM .EQ. 3)PRINT 212
C     REVISE AND THEN DEDUCT STIFFNESS OF FAILED ELEMENTS
      DO 79 I=1,II
C     SHEAR FAILURE
      ICODE=3
      LDA=LA(I)
```

```
      CALL NODES (NDIM,LDA,NOD)
      NDI=NOD(1)
      NDF=NOD(NB-1)
      CALL COORD3(NDI,RDI,TTI,ZZI)
      CALL COORD3(NDF,RDF,TTF,ZZF)
      RO=.5*(RDF+RDI)
      RS=.5*(RDF-RDI)
      ZS=.5*(ZZF-ZZI)
      IF (TTF .LT. TTI)TTF=6.283185307
      TO=.5*(TTF+TTI)
      TS=.5*(TTF-TTI)
      CALL STRESS(LDA,NOD,STRS)
      CALL PSTRES(NDIM,1,STRS,P)
      IF (LRM .NE. 1)GO TO 35
      PT(2)=PD(LDA,1)*TCRT
      PT(3)=.5*(PD(LDA,3)+PD(LDA,1))*SCRT
      GO TO 37
   35 PT(2)=PD(LDA,3)*TCRT
      PT(3)=PD(LDA,3)*SCRT
   37 PT(1)=PD(LDA,3)
      RT1=-P(1)/PD(LDA,3)
      RT2=P(NDIM)/PT(2)
      RT3=.5*(P(NDIM)-P(1))/PT(3)
      IF (RT1 .GT. RT2)GO TO 46
C     TENSILE FAILURE
      IF (RT2 .GT. RT3)ICODE=2
      GO TO 47
C     COMPRESSIVE FAILURE
   46 IF (RT1 .GT. RT3)ICODE=1
   47 PRINT 213,LDA,VV(ICODE),WW(ICODE),(STRS(L1),L1=1,NY),
     1(P(L1),L1=1,NDIM),PT(ICODE)
      IF (MI .EQ. NCYC)GO TO 80
      IF (NER .EQ. 1)GO TO 50
      NN=(LDA-1)/NER+1
      NMN=(NN-1)/MINAM+1
      IDRUM=NMN+12
      NAM=NN-(NMN-1)*MINAM-NM(NMN)
      IF (NAM .EQ. 0)GO TO 52
      DO 48 IR=1,NAM
      READ  (IDRUM)((SL(L1,L2),L1=1,NY),L2=1,NX)
   48 CONTINUE
      NM(NMN)=NN-(NMN-1)*MINAM
      GO TO 52
   50 CALL INTEG(NDIM,LDA,NDI,NDF,SL)
   52 IF (ICODE .NE. 2)GO TO 60
      IF (NTEP .EQ. 1)GO TO 60
      NW=1
      IF (ISO .EQ. 1)GO TO 56
      ISO=1
      E1=PD(LDA,1)
      E2=E1
      IF (NDIM .EQ. 2)E2=0.
      E3=0.
      P1=PD(LDA,2)
      P2=P1
      P3=P1
      G1=E1/2./(1.+P1)
```

```
      G2=G1
      G3=G1
      GO TO 58
C     FOR ORIGINALLY ANISOTROPIC DISCS
   56 CONTINUE
C     ***************
   58 CALL INTEG(NDIM,LDA,NDI,NDF,SLD)
      DO 59 L1=1,NX
      DO 59 L2=1,NX
   59 SLD(L1,L2)=SLD(L1,L2)-SL(L1,L2)
      GO TO 62
   60 DO 61 L1=1,NX
      DO 61 L2=1,NX
   61 SLD(L1,L2)=-POSM*SL(L1,L2)
   62 DO 77 L1=1,NB
      LIL=NOD(L1)
      LX=NOER(LIL)-MX
      MX=NOER(LIL)
      IF (MX .LT. MINUM)MINUM=MX
      IF (LX .LE. 0)GO TO 65
      IF (LX .EQ. 1)GO TO 68
      LX=LX-1
      DO 64 IA=1,LX
      READ  (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
   64 CONTINUE
      GO TO 68
   65 NXX=1-LX
      DO 66 IA=1,NXX
      BACKSPACE 12
   66 CONTINUE
   68 READ  (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
      BACKSPACE 12
      IDUM=(L1-1)*NDIM
      DO 69 IA=1,NDIM
      IIA=IA+IDUM
      DO 69 IB=1,NDIM
      IIB=IB+IDUM
   69 S(IA,IB)=S(IA,IB)+SLD(IIA,IIB)
      IF (MX .EQ. NTNN)GO TO 76
      NN=NN3/NDIM-1
      DO 75 L2=1,NB
      LAL=NOD(L2)
      JX=NOER(LAL)
      IF (JX .LE. MX)GO TO 75
      NBC=NBRC(MX)-1
      DO 71 L3=1,NN
      LL3=L3
      IF (LOC(NBC+L3) .EQ. JX)GO TO 72
   71 CONTINUE
   72 IDEM=LL3*NDIM
      IDAM=(L2-1)*NDIM
      DO 73 IA=1,NDIM
      IIA=IA+IDUM
      DO 73 IB=1,NDIM
      IJA=IB+IDEM
      IIB=IB+IDAM
   73 S(IA,IJA)=S(IA,IJA)+SLD(IIA,IIB)
```

```
   75 CONTINUE
   76 WRITE (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
      BACKSPACE 12
      READ  (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
   77 CONTINUE
      ISO=LRM
   79 CONTINUE
      CALL FCTOR
   80 CONTINUE
  100 FORMAT (915)
  102 FORMAT (8F10.0)
  104 FORMAT (4F10.0,215)
  106 FORMAT (110)
  199 FORMAT (17HOCRITICAL LOAD = ,E12.6)
  200 FORMAT (27HOEXTERNALLY-LOADED NODES --,
     18(14,1H,)/(27X,8(14,1H,)))
  201 FORMAT (18HOBOUNDARY NODES --,10(14,1H,)
     2/(18X,10(14,1H,)))
  202 FORMAT (10HOCYCLE NO.,13,5H ****)
  203 FORMAT (36HOHORIZONTAL STRAINS AT CENTRAL POINT)
  204 FORMAT (17H     FRONT END = ,E10.5)
  205 FORMAT (17H     BACK END  = ,E10.5)
  206 FORMAT (17H     AVERAGE   = ,E10.5)
  207 FORMAT (1H0,11X,'RADIAL DISPLACEMENTS AT LOADED NODES'/
     12(' NODE',5X,' LOAD ',5X,'DISPLACEMENT',10X))
  209 FORMAT (2(14,4X,E10.5,5X,E10.5,10X))
  211 FORMAT (1H0,9X,19HSTRESSES IN FAILED ,
     131HELEMENTS (DUE TO 1000-LB. LOAD))
  213 FORMAT (16,4X,A6,A5,10(1X,E10.5))
  212 FORMAT (' ELEMENT  FAIL. MODE     RADIAL     CIRCUM.   ',
     1'    AXIAL       SHRRC        SHRRZ         SHRCZ    ',
     2'    PRNCP1      PRNCP2       PRNCP3       ALLOW.')
  214 FORMAT (' ELEMENT  FAIL. MODE     RADIAL     CIRCUM.   ',
     1'    SHRRC       PRNCP1       PRNCP2       ALLOW.')
  300 FORMAT (20HIMATERIAL PROPERTIES)
  301 FORMAT (23H      ELASTIC MODULUS = ,1PE12.6)
  302 FORMAT (22H      POISSONS RATIO = ,F4.2)
  303 FORMAT (28H      ALLOW. COMPR. STRESS = ,2PE10.3)
  306 FORMAT (23H      ELASTIC MODULUS = ,1PE12.6,
     14H TO ,1PE12.6)
  307 FORMAT (22H      POISSONS RATIO = ,F4.2,
     14H TO ,F4.2)
  308 FORMAT (28H      ALLOW. COMPR. STRESS = ,2PE10.3,
     14H TO ,2PE10.3)
  309 FORMAT (19H      ELASTIC MODULI/8X,5HE1 = ,1PE12.6/
     18X,5HE2 = ,1PE12.6/8X,5HE3 = ,1PE12.6)
  310 FORMAT (20H      POISSONS RATIOS/8X,5HP1 = ,F4.2/
     18X,5HP2 = ,F4.2/8X,5HP3 = ,F4.2)
  311 FORMAT (17H      SHEAR MODULI/8X,5HG1 = ,1PE12.6/
     18X,5HG2 = ,1PE12.6/8X,5HG3 = ,1PE12.6)
  312 FORMAT (27H      ALLOW. COMPR. STRESSES/8X,5HC1 = ,
     12PE10.3/8X,5HC2 = ,2PE10.3/8X,5HC3 = ,2PE10.3)
  304 FORMAT (28H      ALLOW. TENSL. STRESS = ,F4.2,
     117H OF ALLOW. COMPR.)
  305 FORMAT (28H      ALLOW. SHEAR. STRESS = ,F4.2,
     117H OF ALLOW. COMPR.)
  313 FORMAT (28HODIMENSIONS OF DISC SPECIMEN)
```

```
      314 FORMAT (16H     DIAMETER = ,F5.2)
      315 FORMAT (17H     THICKNESS = ,F5.2)
      316 FORMAT (19HORADIAL COORDINATES/(4X,10(F6.2,1H,)))
      317 FORMAT (38HOCIRCUMFERENTIAL COORDINATES (DEGREES)
     1/(4X,10(F6.2,1H,)))
      318 FORMAT (14HOZ-COORDINATES/(4X,10(F6.2,1H,)))
      319 FORMAT (24HOTOTAL NO. OF ELEMENTS =,I4)
      320 FORMAT (21HOTOTAL NO. OF NODES =,I4)
      326 FORMAT ('     ANISOTROPIC'/8X,'ALPHA1 = ',2PE10.3,
     1' DEGREES'/8X,'ALPHA2 = ',2PE10.3/8X,'ALPHA3 = ',2PE10.3)
      433 STOP
          END

END OF COMPILATION:          NO  DIAGNOSTICS.
```

```
      SUBROUTINE NDIND(LSD,INGA,INGO)
C     **************************************************
C     GENERATES NODES CONNECTED TO NODE I AND WRITES THEM
C                  IN AUXILIARY STORAGE
C     FORMS BRANCH COUNTER ARRAY, NBRC(I)
C     **************************************************
C
      COMMON/RUTH/LOC(25000),NTNN,NB,NP,MISTY
      COMMON/FUNKY/NBRC(500),NREO(500),NOER(500)
      COMMON/BUFFR/NR,NS,NZ,NWOP,IDIM
      DIMENSION NCON(27),LSD(INGA,INGO)
      NT=NS-NWOP
      ND=(NT+1)*(NZ+1)
      M=-1
      DO 16 I=1,NTNN
      M=M+1
      J1=(I-1)/ND
      J2=ND*J1+1
      J3=(I-J2)/(NT+1)
      NCON (2)=I-ND
      NCON (1)=NCON(2)-1
      IF (NWOP .EQ. 1 .AND. 1 .EQ. (J2+J3*NS))NCON(1)=NCON(2)+NT
      NCON (3)=NCON(2)+1
      IF (NWOP .EQ. 1 .AND. I .EQ. I/NS*NS)NCON(3)=NCON(2)-NT
      NCON (4)=NCON(3)+ND
      NCON (5)=NCON(4)+ND
      NCON (6)=I+ND
      NCON (8)=NCON(1)+ND
      NCON (7)=NCON(8)+ND
      IF (IDIM .EQ. 2)GO TO 3
      NCON (9)=NCON(1)-NT-1
      NCON(10)=NCON(2)-NT-1
      NCON(11)=NCON(3)-NT-1
      NCON(12)=NCON(11)+ND
      NCON(13)=NCON(12)+ND
      NCON(14)=NCON(6)-NT-1
      NCON(16)=NCON(9)+ND
      NCON(15)=NCON(16)+ND
      NCON(17)=I-NT-1
      NCON(18)=NCON(1)+NT+1
      NCON(19)=NCON(2)+NT+1
      NCON(20)=NCON(3)+NT+1
      NCON(21)=NCON(20)+ND
      NCON(22)=NCON(21)+ND
      NCON(23)=NCON(6)+NT+1
      NCON(25)=NCON(18)+ND
      NCON(24)=NCON(25)+ND
      NCON(26)=I+NT+1
      IF (J3 .EQ. 0 .OR. J3 .EQ. NZ) GO TO 1
      GO TO 3
    1 NI=(J3/NZ+1)*9
      NF=NI+8
      DO 2 J=NI,NF
    2 NCON(J)=0
    3 IF (J1 .EQ. 0 .OR. J1 .EQ. NR) GO TO 4
```

```
              GO TO 6
            4 NI=1+J1/NR*4
              NF=NI+2
              DO 5 J=NI,NF
              NCON(J)=0
              NCON(J+8)=0
            5 NCON(J+17)=0
            6 IF (NWOP .EQ. 1)GO TO 11
              IF (M .EQ. 0)GO TO 7
              IF (M .EQ. NT) GO TO 8
              GO TO 11
            7 NCON (1)=0
              NCON (8)=0
              NCON (7)=0
              NCON (9)=0
              NCON(16)=0
              NCON(15)=0
              NCON(18)=0
              NCON(25)=0
              NCON(24)=0
              GO TO 11
            8 DO 9 J=3,5
              NCON(J)=0
              NCON(J+8)=0
            9 NCON(J+17)=0
              M=-1
           11 NBRC(I)=26
              IF (IDIM .EQ. 2)NBRC(I)=8
              MM=NBRC(I)
              NCON(MM+1)=0
              MM1=MM-1
              DO 14 J=1,MM1
              IF (NCON(J) .NE. 0) GO TO 14
              J1=J+1
              DO 12 K=J1,MM
              KJ=K
              IF (NCON(K) .NE. 0) GO TO 13
           12 CONTINUE
              NBRC(I)=J-1
              GO TO 15
           13 NCON(J)=NCON(KJ)
              NCON(KJ)=0
           14 CONTINUE
           15 NN=NBRC(I)
              DO 10 J=1,NN
           10 LSD(J,I)=NCON(J)
           16 CONTINUE
              RETURN
              END

END OF COMPILATION:        NO DIAGNOSTICS.
```

```
      SUBROUTINE REORD(LSD,INGA,INGO)
C     ********************************
C     NEAR-OPTIMAL REORDERING OF NODES
C     ********************************
C
      COMMON/RUTH/LOC(25000),NTNN,NB,NP,MISTY
      COMMON/FUNKY/NBRC(500),NREO(500),NOER(500)
      COMMON/INFER/NR,NS,NZ,NWOP,IDIM
      DIMENSION LA(100),LSD(INGA,INGO)
      REWIND 11
      PRINT 100
      J=0
      NZET=0
    6 NDUM=1000
      DO 7 I=1,NTNN
      IF (NBRC(I) .LT. 0) GO TO 7
      IF (NBRC(I) .GE. NDUM) GO TO 7
      NDUM=NBRC(I)
      JLOCT=I
    7 CONTINUE
    8 JLOC=JLOCT
      JLOCT=0
      J=J+1
      NREO(J)=JLOC
      NOER(JLOC)=J
      NN=NBRC(JLOC)
      WRITE (11,200)NN,(LSD(L,JLOC),L=1,NN)
      NZET=NZET+NN
      NM=NN
      IF (J .EQ. NTNN-1)GO TO 24
      DO 23 I=1,NN
      KLOC=LSD(I,JLOC)
      MM=NBRC(KLOC)
      IF (MM .GT. INGA)GO TO 30
      DO 14 JJ=1,MM
      JM=JJ
      IF (LSD(JJ,KLOC) .EQ. JLOC)GO TO 15
   14 CONTINUE
   15 M1=MM-1
      DO 16 LL=JM,M1,1
   16 LSD(LL,KLOC)=LSD(LL+1,KLOC)
      NBRC(KLOC)=NBRC(KLOC)-1
      MM=NBRC(KLOC)
      DO 19 JJ=1,NN
      IF (LSD(JJ,JLOC) .EQ. KLOC)GO TO 19
      DO 18 LL=1,MM
      IF (LSD(JJ,JLOC) .EQ. LSD(LL,KLOC))GO TO 19
   18 CONTINUE
      NBRC(KLOC)=NBRC(KLOC)+1
      M=NBRC(KLOC)
      LSD(M,KLOC)=LSD(JJ,JLOC)
   19 CONTINUE
      IF (NBRC(KLOC) .GT. NM) GO TO 23
      NM=NBRC(KLOC)
      JLOCT=KLOC
```

```
      23 CONTINUE
         NBRC(JLOC)=-NBRC(JLOC)
         IF (JLOCT .EQ. 0) GO TO 6
         GO TO 8
      24 NREO(NTNN)=LSD(1,JLOC)
         LA1=LSD(1,JLOC)
         NOER(LA1)=NTNN
         TNN=NTNN
         ZET=NZET+NTNN
         PRCT=ZET/TNN*100./TNN
         REWIND 11
         NSJ=0
         NTA=NTNN-1
         DO 29 I=1,NTA
         READ (11,200)NN,(LA(L),L=1,NN)
         NBRC(I)=-NN
         NNA=NN-1
         DO 26 J=1,NNA
         LAJ=LA(J)
         J1=J+1
         DO 26 K=J1,NN
         LAK=LA(K)
         IF (NOER(LAK) .GT. NOER(LAJ))GO TO 26
         TEMP=LA(J)
         LA(J)=LA(K)
         LA(K)=TEMP
         LAJ=LA(J)
      26 CONTINUE
         PRINT 102,I,NREO(I),(LA(L),L=1,NN)
         DO 27 K=1,NN
         LAK=LA(K)
         NSJ=NSJ+1
      27 LOC(NSJ)=NOER(LAK)
         NSJ=NSJ+1
         LOC(NSJ)=I
      29 CONTINUE
         PRINT 102,NTNN,NREO(NTNN)
         PRINT 104,PRCT
     100 FORMAT ('1      RESULTS OF NODAL REORDERING'/
        1'0NEW NO.  ORIG. NO.    NODAL CONNECTIONS (ORIG. NOS.)')
     101 FORMAT ('0***** ARRAY LENGTH IS EXCEEDED *****')
     200 FORMAT (32I4)
     102 FORMAT (I6,5X,I6,4X,22I5/(21X,22I5))
     104 FORMAT (32H0PERCENTAGE OF NON-ZERO TERMS = ,F5.2)
         GO TO 31
      30 MISTY=MM
         PRINT 101
      31 RETURN
         END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE ELTOND (IDIM,I,LCON)
C     ******************************************
C     GENERATES ELEMENTS CONNECTED TO NODE I
C     ******************************************
C
C
      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
      DIMENSION LCON(8)
      NT=NS-NWOP
      ND=(NT+1)*(NZ+1)
      NG=NS*NZ
      IF (IDIM .EQ. 2)NG=NS
      J1=(I-1)/ND
      J2=ND*J1+1
      J3=(I-J2)/(NT+1)
      K3=J3
      IF (NWOP .EQ. 1) K3=0
      LCON(1)=I-(ND-NG)*J1-K3
      LCON(2)=LCON(1)-1
      IF (NWOP .EQ. 1 .AND. I .EQ. (J2+J3*NS))LCON(2)=LCON(1)+NT
      LCON(3)=LCON(2)-NG
      LCON(4)=LCON(1)-NG
      IF (IDIM .EQ. 2)GO TO 3
      LCON(5)=LCON(1)-NS
      LCON(6)=LCON(2)-NS
      LCON(7)=LCON(6)-NG
      LCON(8)=LCON(5)-NG
      IF (J3 .EQ. 0 .OR. J3 .EQ. NZ) GO TO 1
      GO TO 3
    1 NI=5-J3/NZ*4
      NF=NI+3
      DO 2 J=NI,NF
    2 LCON(J)=0
    3 IF (J1 .EQ. 0 .OR. J1 .EQ. NR) GO TO 4
      GO TO 6
    4 NI=3-J1/NR*2
      NF=NI+1
      DO 5  J=NI,NF
      LCON(J)=0
    5 LCON(J+4)=0
    6 IF (NWOP .EQ. 1)GO TO 9
      J4=I-J2-J3*(NT+1)
      IF (J4 .EQ. 0) GO TO 7
      IF (J4 .EQ. NT) GO TO 8
      GO TO 9
    7 LCON(2)=0
      LCON(3)=0
      LCON(6)=0
      LCON(7)=0
      GO TO 9
    8 LCON(1)=0
      LCON(4)=0
      LCON(5)=0
      LCON(8)=0
    9 IF (IDIM .EQ. 2)GO TO 10
```

```
      J1=LCON(4)
      LCON(4)=LCON(5)
      LCON(5)=J1
      J1=LCON(3)
      LCON(3)=LCON(6)
      LCON(6)=J1
   10 RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE NODES (IDIM,I,NOD)
C     ******************************************
C     GENERATES NODAL NUMBERS OF         ELEMENT I
C     ******************************************
      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
      DIMENSION NOD(8)
      NT=NS
      IF (NWOP .EQ. 1) NT=NS-1
      ND=(NT+1)*(NZ+1)
      NG=NS*NZ
      IF (IDIM .EQ. 2)NG=NS
      I1=(I-1)/NG
      I2=NG*I1+1
      I3=(I-I2)/NS
      K3=I3
      IF (NWOP .EQ. 1) K3=0
      NOD(1)=I+(ND-NG)*I1+K3
      NOD(2)=NOD(1)+1
      IF (NWOP .EQ. 1 .AND. I .EQ. I/NS*NS)NOD(2)=NOD(1)-NT
      NOD(3)=NOD(2)+ND
      NOD(4)=NOD(1)+ND
      IF (IDIM .EQ. 2)GO TO 10
      NOD(5)=NOD(1)+NT+1
      NOD(6)=NOD(2)+NT+1
      NOD(7)=NOD(6)+ND
      NOD(8)=NOD(5)+ND
   10 RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE COORD3(I,RDC,TTC,ZZC)
C     ***************************************************************
C     ASSIGNS RADIAL,CIRCUMFERENTIAL, AND Z ORDINATES TO NODE I
C     ***************************************************************
      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
      COMMON/LEYTE/RC(30),TC(50),ZC(30)
      ZZC=0.
      NT=NS-NWOP
      ND=(NT+1)*(NZ+1)
      J1=(I-1)/ND
      RDC=RC(J1+1)
      J2=ND*J1+1
      J3=(I-J2)/(NT+1)
      IF (NDIM .EQ. 2)GO TO 1
      ZZC=ZC(J3+1)
    1 J4=I-J2-J3*(NT+1)
      TTC=TC(J4+1)
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE UNIFRM(A,B,C)
      D=RANUN(R)
      C=A+(B-A)*D
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE MTRXB(IDIM,KK,R,TZ,Z,BB)
C     ****************************************************
C     FORMS THE MATRIX B AND THE MATRIX PRODUCT D*B
C     ****************************************************
      COMMON/SANNA/PD(480,3),FNCTR(8),FNCTT(8),FNCTZ(8)
      COMMON/BGAND/C(8,8),B(6,24),M
      COMMON/TANAI/RAD,RS,TO,TS,ZS,NX,HNB
      COMMON/TATAY/ISO,A1,A2,A3,E1,E2,E3,P1,P2,P3,
     1G1,G2,G3
      COMMON/NANAY/TT(3,3),NW
      DIMENSION BB(6,24),FNCT(8),DP(6,6),D(6,6),T(6,6)
      NX1=NX-1
      NY=4*(IDIM-1)*IDIM
      IF (IDIM .EQ. 2)GO TO 3
      FNCT(1)=1./RAD
      FNCT(2)=R/RAD
      FNCT(3)=TZ/RAD
      FNCT(4)=FNCT(3)*R
      FNCT(5)=Z/RAD
      FNCT(6)=FNCT(5)*R
      FNCT(7)=FNCT(5)*TZ
      FNCT(8)=FNCT(7)*R
      FNCTR(2)=1./RS
      FNCTR(4)=TZ/RS
      FNCTR(6)=Z/RS
      FNCTR(8)=FNCTR(6)*TZ
      FNCTT(3)=1./RAD/TS
      FNCTT(4)=R/RAD/TS
      FNCTT(7)=Z/RAD/TS
      FNCTT(8)=FNCTT(7)*R
      FNCTZ(5)=1./ZS
      FNCTZ(6)=R/ZS
      FNCTZ(7)=TZ/ZS
      FNCTZ(8)=FNCTZ(7)*R
      DO 1 I=1,8
      B(1,I)   =FNCTR(I)
      B(2,I)   =FNCT(I)
      B(2,I+8) =FNCTT(I)
      B(3,I+16)=FNCTZ(I)
      B(4,I)   =FNCTT(I)
      B(4,I+8) =FNCTR(I)-FNCT(I)
      B(5,I)   =FNCTZ(I)
      B(5,I+16)=FNCTR(I)
      B(6,I+8) =FNCTZ(I)
    1 B(6,I+16)=FNCTT(I)
      IF (ISO)13,13,6
   13 PR=PD(KK,2)
      D3=.5*PD(KK,1)/(1.+PR)
      DM=D3*2./(1.-2.*PR)
      D1=DM*(1.-PR)
      D2=DM*PR
      DO 2 I=1,24
      BB(1,I)=D1*B(1,I)+D2*(B(2,I)+B(3,I))
      BB(2,I)=D1*B(2,I)+D2*(B(1,I)+B(3,I))
      BB(3,I)=D1*B(3,I)+D2*(B(1,I)+B(2,I))
```

```
      BB(4,1)=D3*B(4,1)
      BB(5,1)=D3*B(5,1)
    2 BB(6,1)=D3*B(6,1)
      GO TO 5
    6 IF (NW .EQ. 1)GO TO 7
      A=A1-T0-TS*TZ
      TT(1,1)=COS(A2)*SIN(A)
      TT(2,1)=-COS(A2)*COS(A)
      TT(3,1)=-SIN(A2)
      TT(1,2)=COS(A3)*COS(A)+SIN(A3)*SIN(A2)*SIN(A)
      TT(2,2)=COS(A3)*SIN(A)-SIN(A3)*SIN(A2)*COS(A)
      TT(3,2)=SIN(A3)*COS(A2)
      TT(1,3)=COS(A3)*SIN(A2)*SIN(A)-SIN(A3)*COS(A)
      TT(2,3)=-COS(A3)*SIN(A2)*COS(A)-SIN(A3)*SIN(A)
      TT(3,3)=COS(A3)*COS(A2)
    7 T(1,1)=TT(1,1)*TT(1,1)
      T(1,2)=TT(1,2)*TT(1,2)
      T(1,3)=TT(1,3)*TT(1,3)
      T(1,4)=2.*TT(1,1)*TT(1,2)
      T(1,5)=2.*TT(1,1)*TT(1,3)
      T(1,6)=2.*TT(1,2)*TT(1,3)
      T(2,1)=TT(2,1)*TT(2,1)
      T(2,2)=TT(2,2)*TT(2,2)
      T(2,3)=TT(2,3)*TT(2,3)
      T(2,4)=2.*TT(2,1)*TT(2,2)
      T(2,5)=2.*TT(2,1)*TT(2,3)
      T(2,6)=2.*TT(2,2)*TT(2,3)
      T(3,1)=TT(3,1)*TT(3,1)
      T(3,2)=TT(3,2)*TT(3,2)
      T(3,3)=TT(3,3)*TT(3,3)
      T(3,4)=2.*TT(3,1)*TT(3,2)
      T(3,5)=2.*TT(3,1)*TT(3,3)
      T(3,6)=2.*TT(3,2)*TT(3,3)
      T(4,1)=TT(1,1)*TT(2,1)
      T(4,2)=TT(1,2)*TT(2,2)
      T(4,3)=TT(1,3)*TT(2,3)
      T(4,4)=TT(1,2)*TT(2,1)+TT(1,1)*TT(2,2)
      T(4,5)=TT(1,3)*TT(2,1)+TT(1,1)*TT(2,3)
      T(4,6)=TT(1,3)*TT(2,2)+TT(1,2)*TT(2,3)
      T(5,1)=TT(1,1)*TT(3,1)
      T(5,2)=TT(1,2)*TT(3,2)
      T(5,3)=TT(1,3)*TT(3,3)
      T(5,4)=TT(1,2)*TT(3,1)+TT(1,1)*TT(3,2)
      T(5,5)=TT(1,3)*TT(3,1)+TT(1,1)*TT(3,3)
      T(5,6)=TT(1,3)*TT(3,2)+TT(1,2)*TT(3,3)
      T(6,1)=TT(2,1)*TT(3,1)
      T(6,2)=TT(2,2)*TT(3,2)
      T(6,3)=TT(2,3)*TT(3,3)
      T(6,4)=TT(2,2)*TT(3,1)+TT(2,1)*TT(3,2)
      T(6,5)=TT(2,3)*TT(3,1)+TT(2,1)*TT(3,3)
      T(6,6)=TT(2,3)*TT(3,2)+TT(2,2)*TT(3,3)
      DUM=E2*E3-E1*E3*P1*P1-E2*E2*P3*P3
     1-E1*E2*(2.*P1*P2*P3+P2*P2)
      D(1,1)=E1*E2*(E3-P3*P3*E2)/DUM
      D(1,2)=E1*E2*(P2*P3*E2+P1*E3)/DUM
      D(1,3)=E1*E2*E3*(P1*P3+P2)/DUM
      D(2,2)=E2*E2*(E3-P2*P2*E1)/DUM
```

```
      D(2,3)=E2*E3*(P1*P2*E1+P3*E2)/DUM
      D(3,3)=E3*E3*(E2-P1*P1*E1)/DUM
      D(2,1)=D(1,2)
      D(3,1)=D(1,3)
      D(3,2)=D(2,3)
      D(4,4)=G1
      D(5,5)=G2
      D(6,6)=G3
      GO TO 11
    3 B(1,2)=1./RS
      B(1,4)=TZ/RS
      B(2,1)=1./RAD
      B(2,2)=R/RAD
      B(2,3)=TZ/RAD
      B(2,4)=B(2,3)*R
      B(2,7)=1./(RAD*TS)
      B(2,8)=B(2,7)*R
      B(3,3)=B(2,7)
      B(3,4)=B(2,8)
      B(3,5)=-B(2,1)
      B(3,6)=1./RS-R/RAD
      B(3,7)=-B(2,3)
      B(3,8)=B(3,6)*TZ
      IF (ISO)8,8,9
    8 PR=PD(KK,2)
      DUM=PD(KK,1)*H/(1.-PR*PR)
      DO 4 I=1,8
      BB(1,I)=DUM*(B(1,I)+PR*B(2,I))
      BB(2,I)=DUM*(B(2,I)+PR*B(1,I))
    4 BB(3,I)=DUM*.5*(1.-PR)*B(3,I)
      GO TO 5
    9 IF (NW .EQ. 1)GO TO 10
      A=AI-TO-TS*TZ
      TT(1,1)=SIN(A)
      TT(1,2)=COS(A)
      TT(2,1)=-COS(A)
      TT(2,2)=SIN(A)
   10 T(1,1)=TT(1,1)*TT(1,1)
      T(1,2)=TT(1,2)*TT(1,2)
      T(1,3)=2.*TT(1,1)*TT(1,2)
      T(2,1)=TT(2,1)*TT(2,1)
      T(2,2)=TT(2,2)*TT(2,2)
      T(2,3)=2.*TT(2,1)*TT(2,2)
      T(3,1)=TT(1,1)*TT(2,1)
      T(3,2)=TT(1,2)*TT(2,2)
      T(3,3)=TT(1,2)*TT(2,1)+TT(1,1)*TT(2,2)
      DUM=E2-P1*P1*E1
      D(1,1)=E1*E2/DUM
      D(1,2)=P1*D(1,1)
      D(2,1)=D(1,2)
      D(2,2)=E2*E2/DUM
      D(3,3)=G1
   11 DO 16 I=1,IDIM
      DO 16 J=1,NX
      DP(I,J)=0.
      DO 17 K=1,IDIM
   17 DP(I,J)=DP(I,J)+D(I,K)*T(J,K)
```

```
   18 CONTINUE
      ID=IDIM+1
      DO 20 I=ID,NX
      DO 20 J=1,NX
   20 DP(I,J)=D(I,I)*T(J,I)
      DO 22 I=1,NX
      DO 22 J=1,NX
      D(I,J)=0.
      DO 21 K=1,NX
   21 D(I,J)=D(I,J)+T(I,K)*DP(K,J)
   22 CONTINUE
      DO 23 I=1,NX1
      I1=I+1
      DO 23 J=I1,NX
   23 D(J,I)=D(I,J)
      DO 25 I=1,NX
      DO 25 J=1,NY
      BB(I,J)=0.
      DO 24 K=1,NX
   24 BB(I,J)=BB(I,J)+D(I,K)*B(K,J)
   25 CONTINUE
    5 RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE INTEG(IDIM,KK,NDI,NDF,SL)
C     **********************************
C     FORMS STIFFNESS MATRIX OF ELEMENTS
C     **********************************
C
      DIMENSION SL(24,24),BB(6,24),AA(2),DX(24,24)
      COMMON/TANA1/RAD,RS,TO,TS,ZS,NX,HNB
      COMMON/RUTH/LOC(25000),NTNN,NB,NP,MINUM
      COMMON/BGAND/C(8,8),B(6,24),H
      DATA AA/.577350269189626,-.577350269189626/
      CALL COORD3(NDI,R1,T1,Z1)
      CALL COORD3(NDF,R2,T2,Z2)
      RS=.5*(R2-R1)
      RO=.5*(R2+R1)
      IF (T2 .LT. T1)T2=6.2831853072
      TS=.5*(T2-T1)
      TO=.5*(T2+T1)
      ZS=.5*(Z2-Z1)
      IF (IDIM .EQ. 2)ZS=1.
      NCX=IDIM-1
      NP1=NP-1
      DO 2 I=1,NP
      DO 2 J=1,NP
    2 SL(I,J)=0.
      DO 7 I=1,2
      R=AA(I)
      RAD=RO+RS*R
      DUM1=RAD*RS*TS*ZS
      DO 6 J=1,2
      T=AA(J)
      DO 5 K=1,NCX
      Z=AA(K)
      CALL MTRXB(IDIM,KK,R,T,Z,BB)
      DO 4 NR=1,NP
      DO 4 NC=NR,NP
      DUM2=0.
      DO 3 N=1,NX
    3 DUM2=DUM2+B(N,NR)*BB(N,NC)
    4 SL(NR,NC)=SL(NR,NC)+DUM1*DUM2
    5 CONTINUE
    6 CONTINUE
    7 CONTINUE
      DO 8 I=1,NP1
      J=I+1
      DO 8 K=J,NP
    8 SL(K,I)=SL(I,K)
      I1=-1
      DO 11 I=1,NP
      I1=I1+1
      I2=(I-1)/IDIM+1
      DO 10 J=1,NP
      DX(I,J)=0.
      DO 9 K=1,NB
      K1=K+I1*NB
    9 DX(I,J)=DX(I,J)+C(K,I2)*SL(K1,J)
```

```
            DX(I,J)=DX(I,J)/HNB
         10 CONTINUE
            IF (I1 .EQ. NCX)I1=-1
         11 CONTINUE
            I1=-1
            DO 14 I=1,NP
            I1=I1+1
            I2=(I-1)/IDIM+1
            DO 13 J=1,NP
            SL(J,I)=0.
            DO 12 K=1,NB
            K1=K+I1*NB
         12 SL(J,I)=SL(J,I)+DX(J,K1)*C(K,I2)
            SL(J,I)=SL(J,I)/HNB
         13 CONTINUE
            IF (I1 .EQ. NCX)I1=-1
         14 CONTINUE
            DO 15 I=1,NP1
            J=I+1
            DO 15 K=J,NP
         15 SL(I,K)=SL(K,I)
            RETURN
            END

END OF COMPILATION:          NO  DIAGNOSTICS.
```

```
          SUBROUTINE STRESS(KK,NOD,STRS)
          COMMON/RUTH/LOC(25000),NTNN,NB,NX,MINUM
          COMMON/TANAI/RO,RS,TO,TS,ZS,NY,HNB
          COMMON/JNFER/NR,NS,NZ,NWOP,IDIM
          COMMON/ADALN/D(3,600)
          COMMON/FUNKY/NBRC(500),NREO(500),NOER(500)
          COMMON/BGAND/C(8,8),B(6,24),H
          DIMENSION NOD(8),DSP(24),A(24),BB(6,24),STRS(6)
C         EVALUATES COMPONENTS OF STRESS ALONG CYLINDRICAL AXES
          II=0
          DO 11 J=1,NB
          NDD=NOD(J)
          MTT=NTNN-NOER(NDD)+1
          DO 10 K=1,IDIM
          II=II+1
       10 DSP(II)=D(K,MTT)
       11 CONTINUE
          II=-1
          DO 15 J=1,NX
          A(J)=0.
          JJ=(J-1)/IDIM+1
          II=II+1
          DO 13 K=1,NB
       13 A(J)=A(J)+C(JJ,K)*DSP(II*NB+K)
          IF (II .EQ. (IDIM-1))II=-1
       15 CONTINUE
          CALL MTRXB(IDIM,KK,0.,0.,0.,BB)
          DO 19 J=1,NY
          STRS(J)=0.
          DO 18 K=1,NX
       18 STRS(J)=STRS(J)+BB(J,K)*A(K)
       19 STRS(J)=STRS(J)/HNB
          RETURN
          END
END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE PSTRES(IDIM,MMA,B,P)
C     ***************************************
C     COMPUTES MAGNITUDES AND DIRECTION COSINES
C           OF PRINCIPAL STRESSES
C     ***************************************
      COMMON/NANAY/CR(3,3),NW
      DIMENSION B(6),P(3)
      IF (IDIM .EQ. 2)GO TO 13
      A1=B(1)+B(2)+B(3)
      A2=B(1)*B(2)+B(2)*B(3)+B(1)*B(3)-
     1B(4)*B(4)-B(5)*B(5)-B(6)*B(6)
      A3=B(1)*B(2)*B(3)+2.*B(4)*B(5)*B(6)-B(1)*B(6)*B(6)
     1-B(2)*B(5)*B(5)-B(3)*B(4)*B(4)
      P(3)=A1/3.
      DM=P(3)
      DO 5 I=1,4
      FX=P(3)*(A2+P(3)*(P(3)-A1))-A3
      FPX=A2+P(3)*(3.*P(3)-2.*A1)
      P(3)=P(3)-FX/FPX
    5 CONTINUE
      PRINT 6,DM,P(3),FX
    6 FORMAT (' FIRST P(3) = ',E10.5,' ** FINAL P(3) = ',
     1E10.5,' ** ITER. ERROR = ',E10.5)
      IF (ABS(FX) .LT. 150.)GO TO 8
      DM=P(3)
      DO 7 I=1,4
      FX=P(3)*(A2+P(3)*(P(3)-A1))-A3
      FPX=A2+P(3)*(3.*P(3)-2.*A1)
      P(3)=P(3)-FX/FPX
    7 CONTINUE
      PRINT 6,DM,P(3),FX
    8 B1=P(3)-A1
      B2=P(3)*B1+A2
      DM=SQRT(B1*B1-4.*B2)
      P(1)=.5*(-B1-DM)
      P(2)=.5*(-B1+DM)
      IF (P(3) .GT. P(2))GO TO 10
      B2=P(2)
      P(2)=P(3)
      P(3)=B2
      IF (P(2) .GT. P(1))GO TO 10
      B1=P(1)
      P(1)=P(2)
      P(2)=B1
   10 IF (MMA .NE. 1)GO TO 20
C     CR = DIRECTION COSINES
      DO 12 I=1,3
      CC=(P(I)-B(2))*(P(I)-B(3))-B(6)*B(6)
      BB=(P(I)-B(1))*(P(I)-B(2))-B(4)*B(4)
      IF (ABS(BB) .GT. ABS(CC))GO TO 11
      AA=(P(I)-B(3))*B(4)+B(5)*B(6)
      BB=(P(I)-B(2))*B(5)+B(4)*B(6)
      GO TO 9
   11 CC=(P(I)-B(2))*B(5)+B(4)*B(6)
      AA=(P(I)-B(1))*B(6)+B(4)*B(5)
```

```
      9 DM=SQRT(CC*CC+BB*BB+AA*AA)
        CR(1,I)=CC/DM
        CR(2,I)=AA/DM
        CR(3,I)=BB/DM
     12 CONTINUE
        GO TO 20
     13 DM=SQRT(.25*(B(1)-B(2))**2+B(3)*B(3))
        P(1)=.5*(B(1)+B(2))-DM
        P(2)=.5*(B(1)+B(2))+DM
        IF (MMA .NE. 1)GO TO 20
        DO 14 I=1,2
        AA=P(I)-B(1)
        BB=P(I)-B(2)
        IF (ABS(BB) .GT. ABS(AA))GO TO 15
        DM=SQRT(AA*AA+B(3)*B(3))
        CR(2,I)=AA/DM
        CR(1,I)=B(3)/DM
        GO TO 14
     15 DM=SQRT(BB*BB+B(3)*B(3))
        CR(1,I)=BB/DM
        CR(2,I)=B(3)/DM
     14 CONTINUE
     20 RETURN
        END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE FACTOR
C     ********************
C     MATRIX FACTORIZATION
C     ********************
      COMMON/FUNKY/NROW(500),NREO(500),NOER(500)
      COMMON/JNFER/NR,NS,NZX,NWOP,IDIM
      COMMON/ABALN/D(3,600)
      COMMON/JANET/NSPD,NWSD(100),SPD(100,3)
      COMMON/BOPAP/NLOD(100),CNLD(100),NE,NER
      COMMON/RUTH/LOC(25000),NTNN,NB,NP,MISTY
      DIMENSION A(3,300),B(3,3),C(3,3),JIT(4),LCON(8),
     1          NOD(8),NSPT(500),S(3,300),SL(24,24)
      EQUIVALENCE (D(1,301),S)
      REWIND 15
      REWIND 14
      REWIND 13
      REWIND 12
      REWIND 11
      REWIND 10
      JIT(1)=0
      JIT(2)=0
      JIT(3)=0
      MIMI=0
      KD=0
      NEE=1
      MEE=1
      NZ=0
      DO 70 I=1,NTNN
      NN=0
      IF (I .EQ. NTNN)GO TO 6
      NN=-NROW(I)
      NROW(I)=NZ+1
      NSPT(I)=NZ+1
      NZ=NZ+NN+1
    6 NN3=(NN+1)*IDIM+1
      DO 7 J=1,IDIM
      DO 7 K=1,NN3
    7 S(J,K)=0.
      J=NREO(I)
      PRINT 100,I,J
  100 FORMAT (6H0NODE ,I4,11H (ORIG. NO.,I4,1H))
      CALL ELTOND (IDIM,J,LCON)
      PRINT 101,(LCON(II),II=1,NB)
  101 FORMAT (34H    ELEMENTS CONNECTED TO NODE -- ,8(I4,1H,))
      DO 15 K=1,NB
      IF (LCON(K) .EQ. 0)GO TO 15
      KK=LCON(K)
      CALL NODES (IDIM,KK,NOD)
      PRINT 102,KK,(NOD(II),II=1,NB)
  102 FORMAT (21H    NODES OF ELEMENT ,I4,4H -- ,8(I4,1H,))
      NDI=NOD(1)
      NDF=NOD(NB-1)
      IF (NER .EQ. 1)GO TO 56
      KS=(KK-1)/NER+1
      IF (KS .EQ. KD)GO TO 57
```

```
      NUM=(KS-1)/MISTY+1
      IDRUM=NUM+12
      JNT=KS-(NUM-1)*MISTY-JIT(NUM)
  399 FORMAT ('0IDRUM=',I3,'***JNT=',I3)
      PRINT 399,IDRUM,JNT
      JIT(NUM)=KS-(NUM-1)*MISTY
      IF (JNT .LE. 0)GO TO 51
      IF (JNT .EQ. 1)GO TO 54
      JNT=JNT-1
      DO 50 IA=1,JNT
      READ  (IDRUM)((SL(L1,L2),L1=1,NP),L2=1,NP)
   50 CONTINUE
      GO TO 54
   51 JMT=1-JNT
      DO 52 IA=1,JMT
      BACKSPACE IDRUM
   52 CONTINUE
   54 READ  (IDRUM)((SL(L1,L2),L1=1,NP),L2=1,NP)
      KD=KS
      GO TO 57
   56 CALL INTEG(IDIM,KK,NDI,NDF,SL)
   57 DO 8 M=1,NB
      IF (J .EQ. NOD(M))GO TO 9
    8 CONTINUE
    9 NDIA=M
      DO 10 KM=1,IDIM
      KN=(NDIA-1)*IDIM+KM
      DO 10 LM=1,IDIM
      LN=(NDIA-1)*IDIM+LM
   10 S(KM,LM)=S(KM,LM)+SL(KN,LN)
      IF (NN .EQ. 0)GO TO 15
      DO 14 M=1,NB
      NODM=NOD(M)
      IF (I .GE. NOER(NODM))GO TO 14
      NRW=NROW(I)-I
      DO 13 L=1,NN
      NRW=NRW+1
      IF (LOC(NRW) .NE. NOER(NODM))GO TO 13
      DO 12 KM=1,IDIM
      KN=(NDIA-1)*IDIM+KM
      DO 12 MM=1,IDIM
      LN=(M-1)*IDIM+MM
      LM=L*IDIM+MM
   12 S(KM,LM)=S(KM,LM)+SL(KN,LN)
      GO TO 14
   13 CONTINUE
   14 CONTINUE
   15 CONTINUE
      IF (J .NE. NLOD(NEE))GO TO 16
      S(1,NN3)=CNLD(NEE)
      NEE=NEE+1
   16 WRITE (12)NN3,((S(L1,L2),L1=1,IDIM),L2=1,NN3)
      IF (MEE .GT. NSPD)GO TO 38
      IF (J .NE. NWSD(MEE))GO TO 18
      DO 17 K=1,IDIM
      IF (SPD(MEE,K) .GE. 100.)GO TO 17
      DO 26 L=1,IDIM
```

```
      S(L,NN3)=S(L,NN3)-S(L,K)*SPD(MEE,K)
      S(K,L)=0.
   26 S(L,K)=0.
      S(K,K)=1.
      S(K,NN3)=SPD(MEE,K)
   17 CONTINUE
      MEE=MEE+1
      IF (MEE .GT. NSPD)GO TO 38
   18 MU=MEE
      NU=1
      NRW1=NROW(I)-1
      DO 28 K=MU,NSPD
      NWS=NWSD(K)
      DO 30 L=NU,NN
      NRW=NRW1+L
      IF (LOC(NRW) .GT. NOER(NWS))GO TO 28
      IF (LOC(NRW) .NE. NOER(NWS))GO TO 30
      MM=L
      M3=(NRW-NROW(I)+1)*IDIM
      DO 39 IA=1,IDIM
      IF (SPD(K,IA) .GE. 100.)GO TO 39
      DO 36 IB=1,IDIM
      S(IB,NN3)=S(IB,NN3)-S(IB,M3+IA)*SPD(K,IA)
      IF (S(IB,IB)-1. .GT. .001)S(IB,M3+IA)=0.
   36 CONTINUE
   39 CONTINUE
      GO TO 37
   30 CONTINUE
      GO TO 38
   37 NU=MM+1
   28 CONTINUE
   38 IF (MIM1 .EQ. 0)GO TO 62
      JILL=1
      II=I-1
      DO 60 K=1,II
      READ  (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      ISPT=NSPT(K)
      IF (LOC(ISPT) .NE. I)GO TO 60
      ITER=(NSPT(K)-NROW(K))*IDIM
      NSPT(K)=NSPT(K)+1
      IF (JILL .GT. NSPD)GO TO 42
      DO 34 LI=JILL,NSPD
      NWD=NWSD(LI)
      KEE=LI
      IF (NOER(NWD) .EQ. K)GO TO 35
      IF (NOER(NWD) .GT. K)GO TO 42
   34 CONTINUE
   35 JILL=KEE+1
      DO 41 IA=1,IDIM
      IF (SPD(KEE,IA) .GE. 100.)GO TO 41
      DO 40 IB=1,IDIM
      S(IB,NN3)=S(IB,NN3)-SPD(KEE,IA)*A(IA,ITER+IB)
   40 A(IA,ITER+IB)=0.
   41 CONTINUE
C     GAUSSIAN ELIMINATION
   42 DO 21 IA=1,IDIM
```

```
      ID=ITER+IA
      DO 21 IB=1,IDIM
      C(IA,IB)=0.
      DO 19 IC=1,IDIM
   19 C(IA,IB)=C(IA,IB)+A(IC,ID)*B(IC,IB)
   21 CONTINUE
      NNO=NNI-ITER
      DO 24 IA=1,IDIM
      DO 23 IB=1,NNO
      D(IA,IB)=0.
      ID=ITER+IB
      DO 22 IC=1,IDIM
   22 D(IA,IB)=D(IA,IB)+C(IA,IC)*A(IC,ID)
      D(IA,IB)=D(IA,IB)/DET
   23 CONTINUE
   24 CONTINUE
      LYN=NNO/IDIM-1
      IF (LYN .EQ. 0)GO TO 32
      NRW=NROW(I)-1
      JS=1
      DO 31 L=1,LYN
      MSPT=ISPT+L
      L3=L*IDIM
      DO 27 M=JS,NN
      NRW=NRW+1
      IF (LOC(MSPT) .NE. LOC(NRW))GO TO 27
      MM=M
      M3=(NRW-NROW(I)+1)*IDIM
      DO 25 IA=1,IDIM
      DO 25 IB=1,IDIM
      IL=L3+IB
      IM=M3+IB
   25 S(IA,IM)=S(IA,IM)-D(IA,IL)
      GO TO 29
   27 CONTINUE
      GO TO 32
   29 JS=MM+1
   31 CONTINUE
   32 DO 33 IA=1,IDIM
      S(IA,NN3)=S(IA,NN3)-D(IA,NNO)
      DO 33 IB=1,IDIM
      ID=ITER+IB
      A(IA,ID)=C(IB,IA)/DET
   33 S(IA,IB)=S(IA,IB)-D(IA,IB)
      BACKSPACE 10
      WRITE (10)NNI,((A(L1,L2),L1=1,IDIM),L2=1,NNI),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      IF (K .EQ. II)GO TO 60
      BACKSPACE 10
      READ  (10)NNI,((A(L1,L2),L1=1,IDIM),L2=1,NNI),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
   60 CONTINUE
   62 DET=S(1,1)*S(2,2)-S(1,2)*S(1,2)
      IF (IDIM .EQ. 3)GO TO 64
      B(1,1)=S(2,2)
      B(2,2)=S(1,1)
      B(1,2)=-S(1,2)
```

```
      GO TO 65
   64 B(1,1)=S(2,2)*S(3,3)-S(2,3)*S(2,3)
      B(1,2)=S(1,3)*S(2,3)-S(1,2)*S(3,3)
      B(1,3)=S(1,2)*S(2,3)-S(1,3)*S(2,2)
      B(2,2)=S(1,1)*S(3,3)-S(1,3)*S(1,3)
      B(2,3)=S(1,2)*S(1,3)-S(1,1)*S(2,3)
      B(3,3)=DET
      DET=S(1,3)*B(1,3)+S(2,3)*B(2,3)+S(3,3)*B(3,3)
      B(3,2)=B(2,3)
      B(3,1)=B(1,3)
   65 B(2,1)=B(1,2)
      IF (I .EQ. NTNN)GO TO 70
      NN1=NN*IDIM+I
      N3=IDIM+1
      WRITE (11)NN1,((S(L1,L2),L1=1,IDIM),L2=N3,NN3)
      WRITE (10)NN1,((S(L1,L2),L1=1,IDIM),L2=N3,NN3),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      IF (MIM1 .EQ. 1)GO TO 69
      DO 66 K=1,I
      ISPT=NSPT(K)
      IF (LOC(ISPT) .EQ. I+1)GO TO 68
   66 CONTINUE
      GO TO 70
   68 MIM1=1
   69 REWIND 10
   70 CONTINUE
C     ******************
C     BACK-SUBSTITUTION
C     ******************
      DO 76 I=1,IDIM
      D(I,1)=0.
      DO 71 J=1,IDIM
   71 D(I,1)=D(I,1)+B(I,J)*S(J,NN3)
   76 D(I,1)=D(I,1)/DET
      DO 75 I=2,NTNN
      NODE=NTNN-I+1
      BACKSPACE 10
      READ  (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      BACKSPACE 10
      DO 77 IA=1,IDIM
      D(IA,I)=0.
      DO 72 IB=1,IDIM
   72 D(IA,I)=D(IA,I)+B(IA,IB)*A(IB,NN1)
   77 D(IA,I)=D(IA,I)/DET
      NRW1=NROW(NODE)-1
      NN=NN1/IDIM
      DO 74 J=1,NN
      JJ=(J-1)*IDIM
      NRW=NRW1+J
      MRW=NTNN-LOC(NRW)+1
      DO 73 IA=1,IDIM
      DO 73 IB=1,IDIM
   73 D(IA,I)=D(IA,I)-A(IA,JJ+IB)*D(IB,MRW)
   74 CONTINUE
   75 CONTINUE
      PRINT 300
```

```
      300 FORMAT ('ONODAL DISPLACEMENTS')
          DO 85 J=1,IDIM
          PRINT 350,(D(J,K),K=NTNN,1,-1)
       85 CONTINUE
      350 FORMAT (12E10.5/(10X,11E10.5))
          RETURN
          END
```

END OF COMPILATION: NO DIAGNOSTICS.

```fortran
      SUBROUTINE FCTOR
      COMMON/FUNKY/NROW(500),NREO(500),NOER(500)
      COMMON/JNFER/NR,NS,NZ,NWOP,IDIM
      COMMON/RUTH/LOC(25000),NTNN,NB,NP,MISTY
      COMMON/JANET/NSPD,NWSD(100),SPD(100,3)
      COMMON/ABALN/D(3,600)
      DIMENSION A(3,300),B(3,3),C(3,3),NSPT(500),S(3,300)
      EQUIVALENCE (D(1,301),S)
      REWIND 10
      REWIND 11
      REWIND 12
      MEE=1
      PRINT 100,MISTY
  100 FORMAT ('0STARTING POINT OF GAUSSIAN ELIMINATION IS ROW',I4)
      IF (MISTY .EQ. 1)GO TO 8
      MRX=MISTY-1
      DO 7 I=1,MRX
      IF (NREO(I) .EQ. NWSD(MEE))MEE=MEE+1
      READ  (12)NN3,((S(L1,L2),L1=1,IDIM),L2=1,NN3)
      READ  (10)NN1,((S(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      READ  (11)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1)
      NN=NN1/IDIM
      NSPT(I)=NROW(I)+NN
      NRW=NROW(I)-1
      DO 5 J=1,NN
      NRW=NRW+1
      IF (LOC(NRW) .LT. MISTY)GO TO 5
      NSPT(I)=NRW
      NNO=(NRW-NROW(I))*IDIM+1
      DO 4 L1=1,IDIM
      DO 4 L2=NNO,NN1
    4 S(L1,L2)=A(L1,L2)
      BACKSPACE 10
      WRITE (10)NN1,((S(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      IF (I .EQ. MRX)GO TO 6
      BACKSPACE 10
      READ  (10)NN1,((S(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      GO TO 6
    5 CONTINUE
    6 PRINT 101,I,NROW(I),NSPT(I)
  101 FORMAT (' I = ',I3,' ** NROW(I) =',I5,' ** NSPT(I) =',I5)
    7 CONTINUE
      REWIND 10
    8 DO 39 I=MISTY,NTNN
      NSPT(I)=NROW(I)
      READ  (12)NN3,((S(L1,L2),L1=1,IDIM),L2=1,NN3)
      NN=NN3/IDIM-1
      IF (MEE .GT. NSPD)GO TO 17
      IF (NREO(I) .NE. NWSD(MEE))GO TO 11
      DO 10 K=1,IDIM
      IF (SPD(MEE,K) .GE. 100.)GO TO 10
      DO 9 L=1,IDIM
```

```
      S(L,NN3)=S(L,NN3)-S(L,K)*SPD(MEE,K)
      S(K,L)=0.
    9 S(L,K)=0.
      S(K,K)=1.
      S(K,NN3)=SPD(MEE,K)
   10 CONTINUE
      MEE=MEE+1
      IF (MEE .GT. NSPD)GO TO 17
   11 NU=1
      NRW1=NROW(I)-1
      DO 16 K=MEE,NSPD
      NWS=NWSD(K)
      DO 14 L=NU,NN
      NRW=NRW1+L
      IF (LOC(NRW) .GT. NOER(NWS))GO TO 16
      IF (LOC(NRW) .NE. NOER(NWS))GO TO 14
      MM=L
      M3=L*IDIM
      DO 13 IA=1,IDIM
      IF (SPD(K,IA) .GE. 100.)GO TO 13
      DO 12 IB=1,IDIM
      S(IB,NN3)=S(IB,NN3)-S(IB,M3+IA)*SPD(K,IA)
      IF (S(IB,IB)-1. .GT. .001)S(IB,M3+IA)=0.
   12 CONTINUE
   13 CONTINUE
      GO TO 15
   14 CONTINUE
      GO TO 17
   15 NU=MM+1
   16 CONTINUE
   17 IF (I .EQ. 1)GO TO 36
      JILL=1
      I1=I-1
      DO 34 K=1,I1
      READ  (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      ISPT=NSPT(K)
      IF (LOC(ISPT) .NE. I)GO TO 34
      ITER=(NSPT(K)-NROW(K))*IDIM
      NSPT(K)=NSPT(K)+1
      IF (JILL .GT. NSPD)GO TO 22
      DO 18 L1=JILL,NSPD
      NWD=NWSD(L1)
      KEE=L1
      IF (NOER(NWD) .EQ. K)GO TO 21
      IF (NOER(NWD) .GT. K)GO TO 22
   18 CONTINUE
   21 JILL=KEE+1
      DO 20 IA=1,IDIM
      IF (SPD(KEE,IA) .GE. 100.)GO TO 20
      DO 19 IB=1,IDIM
      S(IB,NN3)=S(IB,NN3)-SPD(KEE,IA)*A(IA,ITER+IB)
   19 A(IA,ITER+IB)=0.
   20 CONTINUE
C     GAUSSIAN ELIMINATION
   22 DO 25 IA=1,IDIM
      ID=ITER+IA
```

```
      DO 25 IB=1,IDIM
      C(IA,IB)=0.
      DO 23 IC=1,IDIM
   23 C(IA,IB)=C(IA,IB)+A(IC,ID)*B(IC,IB)
   25 CONTINUE
      NNO=NN1-ITER
      DO 27 IA=1,IDIM
      DO 27 IB=1,NNO
      D(IA,IB)=0.
      ID=ITER+IB
      DO 26 IC=1,IDIM
   26 D(IA,IB)=D(IA,IB)+C(IA,IC)*A(IC,ID)
      D(IA,IB)=D(IA,IB)/DET
   27 CONTINUE
      LYN=NNO/IDIM-1
      IF (LYN .EQ. 0)GO TO 32
      NRW=NROW(1)-1
      JS=1
      DO 31 L=1,LYN
      MSPT=ISPT+L
      L3=L*IDIM
      DO 29 M=JS,NN
      NRW=NRW+1
      IF (LOC(MSPT) .NE. LOC(NRW))GO TO 29
      MM=M
      M3=(NRW-NROW(1)+1)*IDIM
      DO 28 IA=1,IDIM
      DO 28 IB=1,IDIM
      IL=L3+IB
      IM=M3+IB
   28 S(IA,IM)=S(IA,IM)-D(IA,IL)
      GO TO 30
   29 CONTINUE
      GO TO 32
   30 JS=MM+1
   31 CONTINUE
   32 DO 33 IA=1,IDIM
      S(IA,NN3)=S(IA,NN3)-D(IA,NNO)
      DO 33 IB=1,IDIM
      ID=ITER+IB
      A(IA,ID)=C(IB,IA)/DET
   33 S(IA,IB)=S(IA,IB)-D(IA,IB)
      BACKSPACE 10
      WRITE (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      IF (K .EQ. I1)GO TO 34
      BACKSPACE 10
      READ  (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
   34 CONTINUE
   36 DET=S(1,1)*S(2,2)-S(1,2)*S(1,2)
      IF (IDIM .EQ. 3)GO TO 64
      B(1,1)=S(2,2)
      B(2,2)=S(1,1)
      B(1,2)=-S(1,2)
      GO TO 65
   64 B(1,1)=S(2,2)*S(3,3)-S(2,3)*S(2,3)
```

```
      B(1,2)=S(1,3)*S(2,3)-S(1,2)*S(3,3)
      B(1,3)=S(1,2)*S(2,3)-S(1,3)*S(2,2)
      B(2,2)=S(1,1)*S(3,3)-S(1,3)*S(1,3)
      B(2,3)=S(1,2)*S(1,3)-S(1,1)*S(2,3)
      B(3,3)=DET
      DET=S(1,3)*B(1,3)+S(2,3)*B(2,3)+S(3,3)*B(3,3)
      B(3,2)=B(2,3)
      B(3,1)=B(1,3)
   65 B(2,1)=B(1,2)
      IF (I .EQ. NTNN)GO TO 39
      NN1=NN*IDIM+1
      N3=IDIM+1
      WRITE (11)NN1,((S(L1,L2),L1=1,IDIM),L2=N3,NN3)
      WRITE (10)NN1,((S(L1,L2),L1=1,IDIM),L2=N3,NN3),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      REWIND 10
   39 CONTINUE
C     BACK-SUBSTITUTION
      DO 76 I=1,IDIM
      D(I,1)=0.
      DO 71 J=1,IDIM
   71 D(I,1)=D(I,1)+B(I,J)*S(J,NN3)
   76 D(I,1)=D(I,1)/DET
      DO 75 I=2,NTNN
      NODE=NTNN-I+1
      BACKSPACE 10
      READ  (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
     1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
      BACKSPACE 10
      DO 77 IA=1,IDIM
      D(IA,1)=0.
      DO 72 IB=1,IDIM
   72 D(IA,1)=D(IA,1)+B(IA,IB)*A(IB,NN1)
   77 D(IA,1)=D(IA,1)/DET
      NRW1=NROW(NODE)-1
      NN=NN1/IDIM
      DO 74 J=1,NN
      JJ=(J-1)*IDIM
      NRW=NRW1+J
      MRW=NTNN-LOC(NRW)+1
      DO 73 IA=1,IDIM
      DO 73 IB=1,IDIM
   73 D(IA,1)=D(IA,1)-A(IA,JJ+IB)*D(IB,MRW)
   74 CONTINUE
   75 CONTINUE
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

APPENDIX B

DATA CARDS PREPARATION GUIDE

The sequential arrangement of the data cards supplied during each program run as well as the input items punched on each card are shown in Table A. Regarding the input parameters, the following explanatory notes are made:

<u>Card # 1</u>

ISØ	-	1 in anisotropic cases; 0 in nonhomogeneous cases; -1 in isotropic cases.
NR	-	Total number of annular surfaces dividing the Brazilian test cylinder.
NS	-	Total number of sectors the circular face of the cylinder or its quadrant is divided into.
NZ	-	Total number of divisions in the Z-direction; 0 in 2-dimensional problems.
NWØP	-	1 if whole circular face of cylinder is involved in analysis; 0 if only a quadrant is involved in analysis.
NDIM	-	2 in 2-dimensional cases; 3 in 3-dimensional cases.
NCYC	-	Total number of load cycles.
NER	-	Least total number of similar and consecutively numbered elements in finite element mesh; 1 in nonhomogeneous and anisotropic cases. This parameter is inserted to avoid having to completely treat each element all over again even if all are the same.
NSS	-	Difference between the numbers of a loaded top node and the diametrically opposite bottom node; 0 if NWØP is 0 or if finite element mesh is symmetric about plane of diametral loads.

Table A Input Cards Arrangement

CARD NUMBER	INPUT DATA	FORMAT
1	ISØ. NR, NS, NZ, NWØP, NDIM, NCYC, NER, NSS	9I5
2*	TCRT, SCRT, EM1, EM2, CS1, CS2, PR1, PR2	8F10.0
3	PØSM, TRN, DIA, H, NTEP, NGØT	4F10.0, 2I5
4	NST1, NST2, MST1, MST2	4I5
5a 5b if needed 5c if needed .	RC(I), I=1, NR1 . . .	8F10.0
6a 6b if needed 6c if needed .	TC(I), J=1, NS1 . . .	8F10.0
7a** 7b if needed 7c if needed .	ZC(I), I=1, NZ1 . . .	8F10.0
8***	N	I10
9****	ALF1, ALF2, ALF3, C1, C2, C3, E1, E2	8F10.0
10****	E3, P1, P2, P3, G1, G2, G3	7F10.0

* Spaces for EM1, EM2, CS1, CS2, PR1, PR2 are left blank in anisotropic cases.
** Supplied in 3-dimensional cases only.
*** Supplied in nonhomogeneous cases only.
****Supplied in anisotropic cases only.

Card # 2

TCRT	-	Ratio of allowable tension to allowable compression.
SCRT	-	Ratio of allowable shear to allowable compression.
EM1, EM2	-	Range of values of elastic moduli.
CS1, CS2	-	Range of values of allowable compressive stresses.
PR1, PR2	-	Range of values of Poisson's ratios.

Card # 3

PØSM	-	Factor to be multiplied to the stiffness matrices of failed elements to obtain that portion of the stiffness matrices to be subtracted from the global stiffness matrix.
TRN	-	Maximum deviation which an element load factor can have from the critical load factor for the element to be considered failed.
DIA	-	Diameter of cylinder.
H	-	Length or thickness of cylinder.
NTEP	-	0 if failure criterion in which the elastic modulus across tension cracks is reduced to zero is to be applied; 1 if the old failure criterion (described in first annual report) is to be applied; always 1 in anisotropic cases.
NGØT	-	0 if both finite element mesh and material property are symmetric about a nodal plane perpendicular to the Z-axis; 1 if no symmetry exists. With this parameter, only one-half of the cylinder length or thickness need be analyzed if symmetry exists.

Card # 4

NST1, NST2 MST1, MST2	-	Pairs of points between which magnitudes of strain, usually the horizontal strain at the center point of each end of the cylinder, are desired.

Card # 5a, etc.

RC(1)	-	0 for solid cylinders; inner radius of hollow cylinders.
RC(2) → RC(NR1)	-	Radial coordinates of annular surfaces.

Card # 6a, etc.

TC(I), I=1, NS1	-	Circumferential coordinates, in degrees, of nodal radial planes measured clockwise from plane of loads. TC(1) = 0.

Card # 7a, etc.

ZC(I), I=1, NZ1	-	Z-coordinates of nodal circles using an end circle as datum. ZC(1) = 0.

Card # 8

N	-	Any 10-digit integer. This is the starting point of random number generator.

Card # 9

ALF1	-	Angle, in degrees, which the line A'B' (see Figure 2.4 of semi-annual report) makes with the plane of the loads.
ALF2, ALF3	-	The angles α_2 and α_3, in degrees, defined in Figure 2.4 of semi-annual report.
	-	These three angles define the directions of anisotropy.

C1, C2, C3 - Allowable compressive stresses in the direction of axes of anisotropy.

E1, E2 - Elastic moduli in the direction of axes of anisotropy, 1,2.

Card # 10

E3 - Elastic modulus in the direction of axis of anistropy 3.

P1, P2, P3 - Poisson's ratios along axes of anisotropy.

G1, G2, G3 - Shear moduli associated with directions of anisotropy.

The entries below the heading FØRMAT in Table A are called format specifications. These serve as instructions to the user on how the input data are to be punched in the data cards. In the present program, each item in the input list has the format specification of the form Fw.d or Iw in which w is the number of column spaces on the card reserved for each item. Given the w's of the entire input list, it is an easy matter to determine the exact columns on the card where each input item is to be punched. For example, in card # 3 one punches PØSM in columns 1-10, TRN in columns 11-20, DIA in columns 21-30, H in columns 31-40, NTEP in columns 41-45, and NGØT in columns 46-50. Integers (items with I specifications) are punched right-justified within the spaces allocated to them. Floating-point numbers (items with F specifications) may be punched with or without the decimal point anywhere within their allocated spaces. If punched without a decimal point, the computer will read the floating-point number as if there was a decimal point d places to the left of the right-most digit. The number before a F or an I is a repetition factor; for instance, (4I5) = (I5, I5, I5, I5).

Input data listings are shown in Chapter 2.